

9-1-1993

Byte After Byte, The Courts Nibble Away at Copyright Protection of Software

Virginia Ann Johnson

Follow this and additional works at: <https://readingroom.law.gsu.edu/gsulr>



Part of the [Law Commons](#)

Recommended Citation

Virginia Ann Johnson, *Byte After Byte, The Courts Nibble Away at Copyright Protection of Software*, 10 GA. ST. U. L. REV. (1993).
Available at: <https://readingroom.law.gsu.edu/gsulr/vol10/iss3/5>

This Article is brought to you for free and open access by the Publications at Reading Room. It has been accepted for inclusion in Georgia State University Law Review by an authorized editor of Reading Room. For more information, please contact mbutler@gsu.edu.

BYTE AFTER BYTE, THE COURTS NIBBLE AWAY AT COPYRIGHT PROTECTION OF SOFTWARE

INTRODUCTION

The Constitution grants Congress the authority to “promote the Progress of Science and useful Arts by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.”¹ To achieve this goal, copyright law grants authors exclusive but limited rights in their original expressions. However, it encourages others to “stand on the shoulders of Giants”² and to advance the ideas and information conveyed in the copyrighted work. Thus, copyright law balances “the interests of authors . . . in the control and exploitation of their writings . . . and society’s competing interest in the free flow of ideas [and] information. . . .”³ However, this balance favors progress, as the constitutional purpose of copyright law is, first, to promote the advancement of science and useful arts, and second, to protect the author’s rights in her creative work.⁴

As technology advances, courts struggle to keep abreast of developments and to understand complex technological issues. In the area of copyright protection of software, the law has evolved sporadically as courts attempt to adapt traditional literary concepts of copyright to unfamiliar, technical issues.⁵ The rules

1. U.S. CONST. art. I, § 8, cl. 8.

2. *Lotus Dev. Corp. v. Paperback Software Int’l*, 740 F. Supp. 37, 77 (D. Mass. 1990) (quoting a letter from Sir Isaac Newton to Robert Hooke (Feb. 5, 1675/1676), “If I have seen further it is by standing on ye shoulders of Giants.”). This notion is reflected in the constitutional mandate to promote the progress of technology by building on the ideas of others.

3. *Sony Corp. of Am. Inc. v. Universal City Studios, Inc.*, 464 U.S. 417, 429-30 (1984).

4. The Supreme Court emphatically stated that “[t]he primary objective of copyright is not to reward the labor of authors, but ‘[t]o promote the Progress of Science and useful Arts.’” *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349 (1991) (quoting U.S. CONST. art. I, § 8, cl. 8). “A prohibition on all copying would stifle the free flow of ideas without serving any legitimate interest of the copyright holder.” *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 843 (Fed. Cir. 1992).

5. “From its beginning, the law of copyright has developed in response to significant changes in technology.” *Sony*, 464 U.S. at 430.

change often and what was permissible yesterday may be illegal tomorrow.

The U.S. Circuit Courts of Appeal differ greatly as to the scope of copyright protection of software, creating a judicial minefield for the unwary. The circuits disagree about what constitutes protected expression, the degree of protection afforded, infringement analysis, and the proper balance between copyright protection and free enterprise. Indeed, a software designer should consider not only how to build on another's ideas without infringing, but where the designer may be sued.⁶

This Note examines some recent significant decisions that limit the scope of copyright protection of software. Section I provides basic software information and definitions. The evolution of copyright protection for software, current statutory rights, and the limitations to those rights are related in Section II. Section III details recent decisions that narrow the judicial scope of protection. Finally, Section IV proposes a new test for copyrightability and substantial similarity, offers suggestions for alternative intellectual property protection, and suggests standards for achieving program compatibility without infringement.

I. SOFTWARE BASICS

The Copyright Act of 1976⁷ defines "computer program" as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."⁸ A programmer writes software or computer programs in a human-readable form called source code.⁹ Source code includes textual

6. If circumstances permit, a plaintiff may engage in forum shopping to take advantage of the intercourt confusion. See *infra* notes 221-45 and accompanying text (*Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990) (holding user interface and menu commands and structure protectable as a whole)). But see *infra* notes 246-60 and accompanying text (*Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006 (N.D. Cal. 1992) (holding menu commands and structure which had become "standardized" not protected by copyright)).

7. Pub. L. No. 94-553, 90 Stat. 2598 (1976) (codified as 17 U.S.C. §§ 101-914 (1988)).

8. 17 U.S.C. § 101 (1988).

9. *Williams Elecs., Inc. v. Artic Int'l, Inc.*, 685 F.2d 870, 876 n.7 (3d Cir. 1982) (quoting NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS (1979) [hereinafter CONTU REPORT]).

references and comments that aid programming and debugging, but are unnecessary to run the computer. The programmer writes the source code in a common alphanumeric computer language such as BASIC, FORTRAN, C, or COBOL.¹⁰

A computer cannot directly use source code and must "compile" the program, converting it to binary form.¹¹ When the program is compiled, unnecessary text such as comments and references is eliminated, leaving only the logical commands and data. The result is called object code.¹² Computer programs are generally sold in object code form, stored in Read Only Memory (ROM) or on a hard drive or floppy disk. Copyright law protects software written in either object or source code,¹³ fixed in any medium and serving any function.¹⁴

A source code is a computer program written in any of several programming languages employed by computer programmers. An object code is the version of a program in which the source code language is converted or translated into the machine language of the computer with which it is to be used.

Id.

10. *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1243 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

11. *Id.* In *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1514 n.2 (9th Cir. 1992), the court explained compilation and decompilation:

A program written in source code is translated into object code using a computer program called an "assembler" or "compiler", and then imprinted onto a silicon chip for commercial distribution. Devices called "disassemblers" or "decompilers" can reverse this process by "reading" the electronic signals for "0" and "1" that are produced while the program is being run, storing the resulting object code in computer memory, and translating the object code into source code.

Id.

12. 714 F.2d at 1243. In *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 836 n.1 (Fed. Cir. 1992), the court offered a clear explanation of object code:

Object code is machine readable, binary code, represented on paper as a series of ones and zeroes. In actuality, those ones and zeroes represent "on" and "off" states of switches on a computer chip . . . [T]he object code, contained in chip memories, is implemented when the chips are operational. When operational, the chips generate a series of "ons" and "offs" in a particular sequence. That results in a pulsating signal which conveys messages to the computer.

Id.

13. *Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1233 (3d Cir. 1986); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 46 (D. Mass. 1990). In fact, a copyright holder need not produce the original source code to prove infringement. *Data Gen. Corp. v. Grumman Sys. Support Corp.*, 803 F. Supp. 487 (D. Mass. 1992).

14. Evan Finkel, *Copyright Protection for Computer Software in the Nineties*, 7

The source and object codes are the literal elements of a computer program. A program's screen displays are also literal elements, and can be protected as a compilation¹⁵ or as audiovisual displays.¹⁶ Copyright law also protects the nonliteral, or unwritten, elements of a program.¹⁷ This nonliteral expression includes the program's user interface, or the "look and feel,"¹⁸ and how the programmer organizes and structures the code.

II. THE EVOLUTION OF COPYRIGHT PROTECTION FOR SOFTWARE AND CURRENT STATUTORY RIGHTS AND LIMITATIONS

A. *Copyright History*

Pursuant to its constitutional grant of power,¹⁹ Congress enacted the Copyright Act of 1909.²⁰ In 1964, the Copyright Office began protecting computer programs as books.²¹ In 1978, Congress overhauled decades of judicial and statutory copyright law by enacting the Copyright Act of 1976 (the 1976 Act or the Copyright Act).²² It provides protection for "original works of authorship fixed in any tangible medium of expression . . . includ[ing] . . . literary works."²³ Congress clearly intended that computer programs are within the scope of "literary works."²⁴

SANTA CLARA COMPUTER & HIGH TECH. L.J. 202, 227 (1991).

15. 17 U.S.C. § 101 (1988). "A 'compilation' is a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship." *Id.*

16. *Id.* § 102(a)(6) (1988). For an example of a successful appeal of Copyright's refusal to register video game as an audiovisual work, see *Atari Games Corp. v. Oman*, 979 F.2d 242 (D.C. Cir. 1992).

17. *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992); *Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir. 1986); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990).

18. A program's "look" includes the choice, sequence, and arrangement of user commands displayed on the computer screen. The "feel" involves the choice of keystrokes and their implementation. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203, 220 (D. Mass. 1992). See Daniel A. Crowe, Note, *The Scope of Copyright Protection for Non-Literal Elements of Computer Software: Computer Associates International, Inc. v. Altai, Inc.*, 37 ST. LOUIS U. L.J. 207, 207 n.7 (1992).

19. U.S. CONST. art. I, § 8, cl. 8.

20. Copyright Act of 1909, ch. 320, 35 Stat. 1075 (1909).

21. Finkel, *supra* note 14, at 202.

22. 17 U.S.C. §§ 101-914 (1988).

23. *Id.* § 102 (1988). Literary works are protected not only in written form, but in any medium that can be "perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device." *Id.*

24. "Literary works . . . include[] . . . computer programs to the extent that they

Congress amended the 1976 Act in 1980, incorporating recommendations from the National Commission on New Technological Uses of Copyrighted Works (CONTU).²⁵ CONTU concluded that intellectual property rights in software would be best protected by copyright law.²⁶ However, CONTU provided no framework with which to analyze infringement of a computer program copyright, so the determination has been left to the courts' ad hoc decisions.²⁷

B. Current Statutory Rights and Limitations

Copyright is not a single right, but a bundle of several exclusive property rights. These include the owner's rights to reproduce the work, prepare derivative works, distribute copies, and import copies into the United States.²⁸ However, Congress created specific limitations to the scope of protection. Copyright protection does not extend to "any idea, procedure, process, system," or method.²⁹ Rather, copyright protection extends to the original expression of the idea, not to the idea itself. This is known as the idea-expression dichotomy, first announced in *Baker v. Selden*,³⁰ now codified at 17 U.S.C. § 102(b). To protect an idea, the design or invention must pass the rigorous investigations and requirements of patent registration.³¹

incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves." H.R. REP. NO. 1476, 94th Cong., 2d Sess. 54 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667.

25. Finkel, *supra* note 14, at 204. Congress established CONTU in 1974 to investigate copyright protection of computer programs. *Id.* at 203.

26. Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1046-47 (1989).

27. "Obviously, no principle can be stated as to when an imitator has gone beyond copying the 'idea,' and has borrowed its 'expression' Decisions must therefore inevitably be ad hoc." *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 704 (2d Cir. 1992) (quoting *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir. 1960)).

28. 17 U.S.C. § 106 (1988) provides, "Subject to sections 107 through 118, the owner of copyright under this title has the exclusive rights to do and to authorize any of the following: (1) to reproduce the copyrighted work in copies . . .; (2) to prepare derivative works based upon the copyrighted work; [and] (3) to distribute copies" *Id.*

29. *Id.* § 102(b) (1988).

30. 101 U.S. 99 (1879).

31. 35 U.S.C. §§ 102-103 (1988). Section 102 requires, among other things, that the invention be novel, not previously patented in the U.S. or abroad, not offered for public use or sale, or not published for more than one year prior to application. *Id.* The invention must be the applicant's invention, unless he has abandoned or sold it.

Otherwise, the idea or information conveyed by the copyright protected expression enters the public domain, where others may build on or advance it.³²

Copyright protection does not extend to all expression, but is limited by several judicially created doctrines. Where the idea and its expression are inseparable, or there are very limited ways to express an idea, the idea and expression have "merged," and copyright protection will not be extended to the expression.³³ To do otherwise would afford a patent-like monopoly on the underlying idea³⁴ for a significantly longer period³⁵ and with much less innovation and investigation than patent registration requires.³⁶ Where idea and expression have "merged," virtually only identical copying will infringe.³⁷

Another limiting doctrine is the *scenes a faire* doctrine, which states that copyright law does not protect expressions that are "indispensable or at least standard"³⁸ to conveying an idea. Thus, an author cannot copyright standard terms or practices. However, when a copyrighted work becomes the de facto industry

Id. Additionally, section 103 requires that the invention be a nonobvious improvement of the previous state of the art. *Id.* Once an idea is patented, the inventor may prohibit others from making, selling, or using the idea. *Id.* § 154 (1988).

32. Harper & Row, Publishers, Inc. v. Nation Enter., 471 U.S. 539, 556-57 (1985).

33. Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971).

34. *Id.*; see Bonito Boats, Inc. v. Thundercraft Boats, Inc., 489 U.S. 141, 159-64 (1989); Sega Enter. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1526 (9th Cir. 1992).

35. Extending the scope of copyright to expression which does not merit protection has significant repercussions. Copyright protection extends for the life of the author and for 50 years after death. 17 U.S.C. § 302(a) (1988). Works made for hire extend for 75 years after publication. *Id.* § 302(c) (1988). In contrast, patent protection lasts for a mere 17 years. 35 U.S.C. § 154 (1988).

36. For example, to acquire patent protection, an innovation must be novel, previously unpublished, and "nonobvious." 35 U.S.C. §§ 102-103 (1988). Patent law's "nonobvious" requirement is a much higher threshold of creativity than the minimal spark required for copyright protection. Philip J. McCabe & William A. Tanenbaum, *Copyright Decisions Increase the Value of Patent Protection for Computer Software*, 5 No. 2 J. PROPRIETARY RTS., Feb. 1993, at 2, 4; see Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 345 (1991); Atari Games Corp. v. Oman, 979 F.2d 242, 243 (D.C. Cir. 1992) ("[T]he requisite level of creativity [for copyrightability] is extremely low."); see also Peter A. Wald et al., *Standards for Interoperability and the Copyright Protection of Computer Programs*, in INTELLECTUAL PROPERTY/ANTITRUST 1992, at 9 (PLI Patents, Copyrights, Trademarks, and Literary Prop. Course Handbook Series No. G4-3885, 1992).

37. Frybarger v. International Bus. Mach. Corp., 812 F.2d 525, 530 (9th Cir. 1987).

38. Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 709 (2d Cir. 1992) (quoting Alexander v. Haley, 460 F. Supp. 40, 45 (S.D.N.Y. 1978)).

standard, the circuits disagree on the scope of protection.³⁹ In addition, external considerations such as compatibility with the computer operating system⁴⁰ with which a program must communicate, computer hardware, and other software may severely constrain the expression a programmer may use.⁴¹ These external factors may render the expression purely functional, and therefore, unprotected.⁴² Finally, expressions in the public domain are not copyrightable.⁴³

Copying that would otherwise violate copyright law does not constitute infringement where it meets the four factors outlined in the fair use statutory exception.⁴⁴ To qualify as a fair use, the court must consider:

- (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
- (2) the nature of the copyrighted work;
- (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
- (4) the effect of the use upon the potential market for or value of the copyrighted work.⁴⁵

The factors are not exclusive, but are "an equitable rule of reason," weighed together to determine if a particular use is a "fair use."⁴⁶

39. See *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992) (holding that industry standardization does not limit copyright protection). But see *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1023 (N.D. Cal. 1992) (holding that functional standard may become an idea, unprotected by copyright law).

40. "An 'operating system' is itself a program that manages the resources of the computer, allocating those resources to the other programs as needed." *Computer Assoc.*, 982 F.2d at 698.

41. *NEC Corp. v. Intel Corp.*, 10 U.S.P.Q.2d 1177, 1188-89 (N.D. Cal. 1989).

42. Functional aspects of a work may be copied. *Baker v. Selden*, 101 U.S. 99, 102-04 (1879), cited in *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992). "[Those] elements of a computer program that are necessarily incidental to its function are similarly unprotectable." *Computer Assoc.*, 982 F.2d at 705.

43. *Computer Assoc.*, 982 F.2d at 710. Section 12 of Pub. L. 100-568 provided that "Title 17 . . . does not provide copyright protection for any work that is in the public domain in the United States." *Id.*

44. 17 U.S.C. § 107 (1988). The exception provides that "fair use of copyrighted work, including such use by reproduction in copies . . . for purposes such as criticism, comment, . . . or research is not an infringement of copyright." *Id.*

45. *Id.*

46. *Sega*, 977 F.2d at 1522 (quoting *Harper & Row, Publishers, Inc. v. Nation Enter.*, 471 U.S. 539, 560 (1985)).

C. Copyright Infringement

Not all copying is infringement.⁴⁷ To prove infringement, a plaintiff must show that the expression copied is protected by copyright and that the copying was unlawful.⁴⁸ A plaintiff may prove unlawful copying either with direct evidence of copying or by showing that the defendant had access to the program together with a substantial similarity between the two works.⁴⁹ Because the statute provides little guidance regarding standards for determining infringement,⁵⁰ the courts must rely on case law, however inconsistent it may be.⁵¹

D. Standards for Determining Copyright Infringement

Most copyright infringement is proven indirectly, with evidence of substantial similarity between two works. Although each circuit performs its own test for determining infringement,⁵² many of the tests use two basic steps. First, the court analyzes copyrightability, or the scope of copyright protection afforded the work.⁵³ Then, the court compares the protected elements or arrangement to the allegedly infringing work to decide if they are substantially similar. This step determines the scope of illicit copying.⁵⁴

47. Indeed, the Supreme Court has acknowledged that an author's labor may be used by others without compensation in some cases. *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349 (1991).

48. *Id.* at 1296; *Computer Assoc.*, 982 F.2d at 701; see 17 U.S.C. § 501 (1988).

49. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 837-38 (Fed. Cir. 1992); *Computer Assoc.*, 982 F.2d at 701.

50. 17 U.S.C. § 501(a) (1988) provides that "[a]nyone who violates any of the exclusive rights of the copyright owner as provided by sections 106 through 118 . . . is an infringer of the copyright."

51. See *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992) (holding that the "look and feel" of a computer program is protectable expression). *But see Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1020 (N.D. Cal. 1992) (holding that "look and feel" is not protected unless each element merits protection).

52. See *Computer Assoc.*, 982 F.2d at 693 (abstraction-filtration-comparison test); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1243 (3d Cir. 1983) (extrinsic-intrinsic analysis); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992) (copyrightability and substantial similarity tests).

53. *Computer Assoc.*, 982 F.2d at 693 (abstraction-filtration components); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. 78 (D. Mass. 1992) (copyrightability component).

54. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. at 97.

Abstraction,⁵⁵ the method of discerning idea from expression, has been a source of confusion for the courts. While some courts have considered the program as expression conveying a single idea,⁵⁶ other courts have recognized that a program is composed of many ideas. Such courts have divided the program into discrete tasks, then determined what is protected expression. This "dissection" method more effectively protects the elements that deserve protection, but may narrow the scope of protection for the program as a whole.⁵⁷ Other courts have analyzed the program as a whole, because protection of the nonliteral elements such as the structure, sequence, and organization (SSO) and the "look and feel" may suffer more under the dissection analysis.⁵⁸ A program's SSO and its "look and feel" are holistic nonliteral expressions that necessarily encompass some unprotected material.⁵⁹

Finally, some courts compare only the copyright protected expression to the infringing work to determine if a substantial amount of the protected expression has been copied.⁶⁰ Other courts compare all similarities, then determine if most result from copying protected expression.⁶¹ These comparisons determine whether the two works are substantially similar.⁶²

III. RECENT DECISIONS ERODE SOFTWARE COPYRIGHT PROTECTION

Several recent decisions have eroded the copyright protection of software.⁶³ The circuits are conducting more detailed analyses

55. *Computer Assoc.*, 982 F.2d at 706.

56. *Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1236 (3d Cir. 1986).

57. *Computer Assoc.*, 982 F.2d at 712.

58. "The mere fact that a work is copyrighted does not mean that every element of the work may be protected." *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 348 (1991); see *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992).

59. See *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 67 (D. Mass. 1990).

60. *Computer Assoc.*, 982 F.2d at 710-11.

61. See *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006 (N.D. Cal. 1992).

62. Finkel, *supra* note 14, at 221.

63. See *Feist Publications, Inc.*, 499 U.S. at 259-60 (overruling the "sweat of the brow" doctrine: effort alone does not confer copyright protection); *Computer Assoc.*, 982 F.2d at 721 (holding that structure, sequence, and organization elements not protected unless entirely within the ambit of copyright protection); *Sega Enter. Ltd. v. Accolade*,

of copyright infringement and generally employing expert testimony to clarify the complex, technological issues.⁶⁴ This closer scrutiny of expression narrows the scope of copyright protection.⁶⁵ Thus, most courts are finding infringement only where it truly exists.

Software copyright litigation falls into three primary categories: alleged copying of the literal code, the nonliteral elements of the code, and the screen displays and user interface.⁶⁶ This section contrasts some past and recent decisions that show the courts narrowing the scope of copyright protection for both literal and nonliteral expression. There are some exceptions to the apparent trend, however.⁶⁷

A. Protection of Literal Expression: The Legality of Reverse Engineering

1. Reverse Engineering in a Nutshell

One common area of litigation involves reverse engineering of the object code, usually stored in ROM. Reverse engineering is the process of obtaining the object code and working backward to generate a version of the source code in order to understand how a program works.⁶⁸ Most reverse engineering is done by reading the ROM electronically to obtain the object code, which is then "decompiled" or "disassembled" into assembly language,⁶⁹ printed out and analyzed.⁷⁰ The analysis reveals the reverse engineer's interpretation of the program's basic structure and functions.⁷¹

Inc., 977 F.2d 1510, 1514 (9th Cir. 1992) (holding that reverse engineering is a "fair use" if the unprotected concepts are otherwise inaccessible); *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1020 (N.D. Cal. 1992) (holding that "look & feel" not protected unless each element merits protection).

64. See *infra* notes 151-55, 174, 212-13, 290-91 and accompanying text.

65. *Computer Assoc.*, 982 F.2d at 712.

66. Crowe, *supra* note 18, at 209.

67. See *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990). Both *Lotus* cases expand copyright protection for a program's "look and feel."

68. *Sega Enter. Ltd. v. Accolade, Inc.*, 785 F. Supp. 1392, 1394-95 (N.D. Cal. 1992).

69. Assembly language is an alphanumeric code which is intelligible to computer programmers, but lacks the textual content of source code. It is essentially a form of shorthand representing the source code. See *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 844 n.6 (Fed. Cir. 1992).

70. *Id.*

71. Brief for Accolade at 85, *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th

Two recent cases have directly addressed the issue of whether the intermediate copying essential to reverse engineering violates copyright protection of the object code stored in ROM.⁷² Past cases have dealt with the issue indirectly, but their holdings have turned on the substantial similarity of the final products.⁷³

2. *Sega v. Accolade: Round 1—The District Court Decision*

In *Sega Enterprises Ltd. v. Accolade, Inc.*,⁷⁴ the District Court for the Northern District of California held that Sega was likely to prevail on a copyright infringement suit against Accolade and granted Sega a preliminary injunction against Accolade. The court ruled that Accolade most likely infringed on Sega's copyrighted object code because it made intermediate copies of the code during the reverse engineering process.⁷⁵

Sega manufactures "Genesis," a home video entertainment system that uses video game cartridges, supplied by Sega and their licensees. Accolade also manufactures video game cartridges, which are compatible to several video game consoles, including Genesis.⁷⁶

In 1989, Accolade first attempted to create video games that would be compatible with the Sega Genesis console.⁷⁷ Accolade purchased Sega game cartridges, successfully reverse engineered Sega's object code, and ascertained the code's functions. Then using Sega's functional parameters only, Accolade developed its

Cir. 1992) (No. 92-15655).

72. *Sega*, 977 F.2d at 1510; *Atari*, 975 F.2d at 832. Both cases held that reverse engineering to access ideas stored in object code is legal in specific cases.

73. In other cases involving reverse engineering, the decisions turned on whether the final product showed substantial similarity to the allegedly infringed work. *Sega*, 977 F.2d at 1519. In *NEC Corp. v. Intel Corp.*, 10 U.S.P.Q.2d 1177 (N.D. Cal. 1989), the court assumed that intermediate copying had occurred during the course of reverse engineering, but held that the final products (micro codes) were not substantially similar, and thus, no infringement was found. Also, the court in *E.F. Johnson Co. v. Uniden Corp. of Am.*, 623 F. Supp. 1485 (D. Minn. 1985), held that even though intermediate copying occurred in the reading and analysis of object code, that alone did not constitute infringement. *Id.* at 1501 n.17; see also *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173 (9th Cir. 1989).

74. *Sega Enter. Ltd. v. Accolade, Inc.*, 785 F. Supp. 1392 (N.D. Cal. 1992).

75. *Id.* at 1402.

76. *Id.* at 1394.

77. *Id.* at 1395.

own code to be Genesis-compatible.⁷⁸ Accolade began marketing Genesis-compatible video games in 1990.⁷⁹

In response to Accolade and other competitors, Sega developed a new console, the Genesis III, which included a licensed trademark security system (TMSS).⁸⁰ TMSS contained a lockout code that the Genesis III read and permitted the display of only Sega-licensed games. To ascertain the lockout code, Accolade disassembled and copied more Sega game cartridges. Accolade's programmers found the code that it thought to be the "key" to the lockout scheme, and incorporated the appropriated code into its newest game cartridges.⁸¹ As a result, Sega sued to enjoin Accolade from further reverse engineering, modifying or embellishing Sega's object code, or developing or manufacturing any game cartridge based on the reverse engineered code.⁸²

The district court held that Sega was entitled to a preliminary injunction because Sega had demonstrated a likelihood of success on the merits of its infringement case against Accolade. The court, interpreting the text of the 1976 Act literally, reasoned that Accolade had infringed on Sega's exclusive rights to reproduce the object code and make derivative works because it had made intermediate copies of Sega's code during reverse engineering.⁸³

78. *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1514-15 (9th Cir. 1992).

79. *Sega Enter. Ltd. v. Accolade, Inc.*, 785 F. Supp. 1392, 1395 (N.D. Cal. 1992).

80. *Id.* When a Sega video game was engaged in the Genesis III console, TMSS caused the message, "PRODUCED BY OR UNDER LICENSE FROM SEGA ENTERPRISES LTD," to be displayed on the video screen. This message was devised primarily to prosecute Taiwanese counterfeiters because Taiwan allows trademark pirates to be prosecuted, but does not recognize foreign copyrights. *Sega*, 977 F.2d at 1515. By incorporating the TMSS code into their code, Accolade's games caused the Sega message to appear on the user's video screen. *Id.* Because Accolade is not a Sega licensee, Sega also sued Accolade for trademark infringement. *Id.* at 1516. The trademark infringement issue is beyond the scope of this Note.

81. *Sega*, 785 F. Supp. at 1395. Accolade appropriated about 20 to 25 bytes of Sega's code to defeat the TMSS. *Sega*, 977 F.2d at 1516. Each video game contains 500,000 to 1,500,000 bytes of code. Sega did not raise the issue of copyright infringement of this appropriated code on appeal, but reserved the right to prosecute Accolade. *Id.* at 1528.

82. *Sega*, 785 F. Supp. at 1395. Sega's claims against Accolade were based on both illegal reproduction and preparation of derivative works via the appropriated object code. *Id.* To obtain the preliminary injunction, Sega needed only to demonstrate "likelihood of success on the merits and the possibility of irreparable injury." *Id.* (quoting *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1174 (9th Cir. 1989)).

83. 785 F. Supp. at 1396; see *supra* note 28.

Accolade argued that intermediate copying could not be the basis for infringement because the end products were not substantially similar to Sega's video games.⁸⁴ The court stated that it need not reach the question of substantial similarity in the completed works because the intermediate copying itself was actionable infringement.⁸⁵ The court noted that the text of the 1976 Act did not limit a finding of infringement to a comparison of final products.⁸⁶

The court relied heavily on *Walt Disney Productions v. Filmmation Associates*,⁸⁷ which held that a copy is an intelligible expression that is fixed in some permanent form that can be perceived directly or with the aid of a machine.⁸⁸ In *Sega*, the district court held that the copies of the object code made by Accolade were fixed works that constituted copies because they could be perceived visually (paper printouts) or communicated via electronic means (a computer).⁸⁹ Further, the court held that any copy prepared as a fixed iteration or a different version and capable of tangible expression constituted an infringing copy, regardless of the fact that it was not a final product.⁹⁰ The violations of the copyright owner's exclusive rights to control the making of reproductions and derivative copies were actionable infringement. The infringing copy need not also infringe on the copyright holder's exclusive rights to distribute the holder's work, as would an infringing final product offered for sale.⁹¹ Thus,

84. 785 F. Supp. at 1396.

85. *Id.* at 1397.

86. *Id.* at 1396; see also 17 U.S.C. § 107 (1988).

87. 628 F. Supp. 871 (C.D. Cal. 1986).

88. *Id.* "To constitute an actionable copy, therefore, an expression need only be permanently case[d] [sic] in some intelligible form." *Id.* at 876, quoted in *Sega*, 785 F. Supp. at 1396; 17 U.S.C. § 101 (1988) provides that

"Copies" are material objects . . . in which a work is fixed by any method now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device A work is "fixed" in a tangible medium of expression when its embodiment in a copy . . . is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration.

Id.

89. *Sega*, 785 F. Supp. at 1396.

90. *Id.* at 1397; Walker v. Univ. Books, Inc. 602 F.2d 859 (9th Cir. 1979); see *Walt Disney*, 628 F. Supp. at 876, quoted in *Sega*, 785 F. Supp. at 1396. ("Where a work is prepared over a period of time, the portion of it that has been fixed at any particular time constitutes the work as of that time, and where the work has been prepared in different versions, each version constitutes a separate work.")

91. Sony Corp. of Am. Inc. v. Universal City Studios, Inc., 464 U.S. 417, 474 (1984)

Accolade violated Sega's copyright by creating copies of Sega's object code while electronically reading, disassembling, recording, and evaluating the code, so the court need not have addressed substantial similarity at all.⁹²

Accolade also raised the statutory defense of "fair use."⁹³ However, the court determined that Accolade's purpose for the use was for financial gain that would directly and adversely affect Sega's sales, which is presumptively not a "fair use."⁹⁴ Moreover, because the nature of Sega's object code was an "unpublished work," it was subject to a "narrower scope of fair use."⁹⁵ The third and fourth elements weighed against Accolade because the entire code was copied for a competitive purpose, which would adversely affect the market for Sega's copyrighted games.⁹⁶ Therefore, the district court held that Accolade's copying was not a fair use.⁹⁷

Finally, Accolade argued that reversing the engineering and the requisite immediate copying should be allowed as a matter of public policy.⁹⁸ The electronics industry relies heavily on reverse engineering as the principal way to study object code and develop compatible or interoperable products.⁹⁹ The court rejected this argument, stating that there was no evidence of legislative intent to allow reverse engineering and intermediate copying as excepted copying.¹⁰⁰ Thus, the court concluded that Sega was likely to succeed on the merits of its claim of copyright

(Blackmun, J., dissenting); *Sega*, 785 F. Supp. at 1397; *Walt Disney*, 628 F. Supp. at 876; see *supra* note 28.

92. *Sega*, 785 F. Supp. at 1397; see *supra* notes 48-49 and accompanying text.

93. 17 U.S.C. § 107 (1988); see *supra* notes 44-46 and accompanying text.

94. *Sony Corp.*, 464 U.S. at 451; see *Harper & Row, Publishers, Inc. v. Nation Enter.*, 471 U.S. 540, 566-67 (1985).

95. *Harper & Row*, 471 U.S. at 564.

96. *Sega*, 785 F. Supp. at 1398.

97. *Id.* at 1398-99.

98. *Id.* at 1399.

99. *Id.*

100. *Id.* The court stated that Congress did not intend to make software reverse engineering a "fair use" because Congress specifically provided a statutory exception for reverse engineering of silicon chip mask works in the Semiconductor Chip Protection Act (SCPA) and chose not to amend the Copyright Act to provide for reverse engineering as a copyright "fair use." 17 U.S.C. § 906 (1988). However, in enacting the SCPA, "Congress expressly stated that it did not intend to limit, enlarge or otherwise affect . . . copyright protection . . . in computer programs . . ." *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1521 (9th Cir. 1992) (quoting H.R. REP. NO. 781, 98th Cong., 2d Sess. at 28, reprinted in 1984 U.S.C.C.A.N. 5750, 5777).

infringement and issued a preliminary injunction against Accolade.¹⁰¹

3. *Sega v. Accolade: Round 2—The Ninth Circuit Speaks*

On appeal, the Ninth Circuit Court of Appeals lifted the preliminary injunction, reversed in part the lower court's decision, and held that reverse engineering was a fair use, subject to specific limitations.¹⁰² The court held that

persons who are neither copyright holders nor licensees [may] disassemble a copyrighted computer program in order to gain an understanding of the unprotected functional copyright elements of the program . . . [w]hen the person seeking the understanding has a legitimate reason for doing so and when no other means of access to the unprotected elements exists.¹⁰³

This holding limited the assertion of the fair use defense for reverse engineering in three significant ways. First, the reverse engineer must seek access to the unprotected ideas and expression contained in the object code.¹⁰⁴ Because much of the code may be unprotected expression dictated by function, efficiency, and external demands,¹⁰⁵ primarily utilitarian works such as computer programs merit weak copyright protection.¹⁰⁶ Further, because the programs are distributed in incomprehensible electronic form,¹⁰⁷ the reverse engineer cannot selectively copy and analyze only unprotected ideas and expression, but must disassemble, copy, and analyze the entire code.¹⁰⁸

Second, the reverse engineer must have a legitimate reason to copy and study the copyrighted code.¹⁰⁹ Accolade's purpose in copying Sega's code was to study its functions and create Genesis-compatible video games.¹¹⁰ Their purpose of building

101. *Sega*, 785 F. Supp. at 1394.

102. *Sega*, 977 F.2d at 1514. The trademark issues raised on appeal are beyond the scope of this Note.

103. *Id.*

104. *Id.*

105. *Id.* at 1524; see *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 1253-56 (2d Cir. 1992); see *supra* notes 40-42 and accompanying text.

106. *Sega*, 977 F.2d at 1527.

107. See *supra* notes 11-14 and accompanying text.

108. *Sega*, 977 F.2d at 1525-26.

109. *Id.* at 1514.

110. *Id.* at 1522. Accolade's actions bolstered its claim that it sought merely to

on Sega's ideas is precisely the growth in arts and sciences decreed by the Constitution.¹¹¹ This scheme also prevents duplicative, wasteful efforts by allowing ideas to enter the public domain and serve as the catalyst for innovations.¹¹² However, copying for purely commercial exploitation weighs against the reverse engineer in the fair use analysis.¹¹³

Finally, the reverse engineer must have no other means of access to the protected code.¹¹⁴ In traditional literary copyright cases, copying is not necessary because the ideas and expression are easily accessed visually. In *Sega*, the Ninth Circuit recognized the unique incomprehensibility of object code stored electronically and the fact that copying is often necessary to understand the underlying concepts.¹¹⁵ Where the unprotected elements can be discovered visually, as in the screen displays, reverse engineering is not a fair use.¹¹⁶ Although the object code, composed of millions of ones and zeroes, can be read visually, no human could read and comprehend the entire code without recording (copying) it.¹¹⁷ Because reverse engineering

study Sega's program functions. *Id.* Using only the functional parameters ascertained from the disassembly analysis, Accolade created its own original games. *Id.* It did not slavishly copy and incorporate Sega's code into its own games, with the exception of 20 to 25 bytes of TMSS code. *Id.* at 1510; cf. Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 HARV. L. REV. 978, 1019 (1993) (arguing that *Sega* is a disguised argument for standardization).

111. *Sega*, 977 F.2d at 1523; cf. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 69 (D. Mass. 1990) ("[T]he desire to achieve 'compatibility' or 'standardization' cannot override the rights of authors to a limited monopoly in the expression embodied in their intellectual 'work.'").

112. *Sega*, 977 F.2d at 1527 (citing *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349-50 (1991); *Rosemont Enter., Inc. v. Random House, Inc.*, 366 F.2d 303, 310 (2d Cir. 1966), *cert. denied*, 305 U.S. 1009 (1967)). The entry of ideas into the public domain is limited by patent law protection. 35 U.S.C. § 101 (1988).

113. This presumption can be rebutted by the nature of the commercial use. *Sega*, 977 F.2d at 1522; see *Hustler Magazine, Inc. v. Moral Majority, Inc.*, 796 F.2d 1148, 1152 (9th Cir. 1986).

114. *Sega*, 977 F.2d at 1514.

115. *Id.* at 1525.

116. *Id.* at 1520. "In reality, however, viewing how a program, such as a word processor, operates through its screen displays tells a programmer virtually nothing about the ideas or functional elements underlying the source code." Stuart D. Levi, "Look and Feel" Meets "SSO"—How Courts Are Confusing The Relationship Between Computer Screen Displays and Source Code, 5 No. 1 J. PROPRIETARY RTS., Jan. 1993, at 2, 6. Thus, the Ninth Circuit does not comprehend the relationship between screen displays and the underlying code. *Id.* The district court in *Manufacturers Technologies, Inc. v. CAMS, Inc.*, 706 F. Supp. 984 (D. Conn. 1989), recognized the distinctions. Levi, *supra*, at 8-9; see *infra* notes 292-94 and accompanying text.

117. *Sega*, 977 F.2d at 1525. To understand the ideas contained in the object code,

was the only means to understand the underlying ideas in Sega's object code, Accolade could assert the fair use defense.

Accolade raised four arguments on appeal in defense of the infringement charges. First, it asserted that intermediate copying does not infringe on Sega's copyright because the end products were not substantially similar. On this point, the Ninth Circuit affirmed the district court's holding that neither the language of the 1976 Act nor Ninth Circuit case law limited infringement to use of a copy in an end product.¹¹⁸ Accolade's reverse engineering that generated unauthorized copies was infringement.¹¹⁹

Second, Accolade argued that because disassembly is necessary to access the unprotected ideas and expression in the object code, it should not be infringement. The court rejected this contention, stating that case law clearly holds that copyright law protects object code, regardless of how embodied¹²⁰ or how inaccessible to humans.¹²¹ The need to disassemble object code arises only when no other avenue exists to comprehend the program's functions.¹²² Therefore, infringement cases arising from disassembly should be evaluated ad hoc using an equitable fair use analysis.¹²³

Third, Accolade argued that section 117 of the Copyright Act authorized a user to copy a program if making the copy is a step

by itself unintelligible to humans, the code must be translated from binary form to human-readable form. *Id.* It is impossible for a human to read and remember more than a few bytes at a time. *Id.* Thus, the code must be electronically read, disassembled, and converted to assembly language. *Id.* Programmers then study the resulting code and interpret how the program works. *Id.* Because the comments and textual references originally used in the source code are not utilized in the object code after compilation, the process of reverse engineering cannot reconstruct that text. *Id.* The final product is not a copy of the original source code, but a reverse engineer's interpretation of how the program works. Brief for Accolade at 14-17, *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992) (No. 92-15655).

118. *Sega*, 977 F.2d at 1518-19. The 1976 Act clearly grants to the copyright holder the exclusive rights to reproduce and transform his own work and to authorize others to do the same. *Id.* at 1518. Anyone who violates these exclusive rights has infringed. *Id.* Therefore, the Copyright Act prohibits intermediate copying. *Id.* at 1519.

119. *Id.*

120. *Id.* at 1519-20; see *supra* note 13-14.

121. *Sega*, 977 F.2d at 1519; *Apple Computer, Inc. v. Formula Int'l, Inc.*, 725 F.2d 521, 525 (9th Cir. 1984); see *Miller, supra* note 110, at 988-90 (arguing that access does not justify limiting copyright protection).

122. *Sega*, 977 F.2d at 1520.

123. *Id.*

required to use the program.¹²⁴ However, Congress enacted section 117 to protect the ordinary software consumer from copyright infringement when copying a program into a personal computer.¹²⁵ Accolade's argument vastly distorted the congressional intent of section 117.

Finally, as discussed above, the court agreed that Accolade's intermediate copying was a fair use exception to copyright infringement.¹²⁶ The first fair use factor considers the purpose of the use, including the commercial or nonprofit characteristics.¹²⁷ Sega characterized Accolade's use as commercial exploitation to develop competitive video games.¹²⁸ Commercial use weighs against the infringer in a fair use analysis.¹²⁹ However, the court found that this first factor weighed in favor of Accolade because its primary purpose of disassembling the code for analysis was a legitimate one and its commercial use was indirect.¹³⁰

However, the court weighed Accolade's eventual, if indirect, commercial purpose for disassembling Sega's code in analyzing the fourth statutory factor, the effect on the market for the copyrighted work.¹³¹ Usurping the market for a copyrighted work, or diminishing its sales, can make this factor the deciding one.¹³² The Ninth Circuit noted that typical video game consumers buy multiple games and that Accolade's dissimilar video games would not directly affect the market for Sega's games.¹³³ A copyright holder cannot use the Copyright Act to

124. *Id.*

125. *Id.* (citing CONTU REPORT, *supra* note 9, at 13).

126. *Sega*, 977 F.2d at 1514, 1520. Sega had contended that Congress intended 17 U.S.C. § 117 to be the only authorized copying permissible under the statute, such that any other copying cannot be considered a fair use. *Id.* at 1520-21. However, as the court noted, "sections 107 and 117 serve entirely different functions. Section 117 defines a narrow category of copying that is *lawful per se*. 17 U.S.C. § 117. Section 107, by contrast, establishes a *defense* to an otherwise valid claim of copyright infringement." *Sega*, 977 F.2d at 1521.

127. 17 U.S.C. § 107(1) (1988).

128. *Sega*, 977 F.2d at 1522.

129. *Id.*; see *supra* notes 44-46; *Sega Enter. Ltd. v. Accolade, Inc.*, 785 F. Supp. 1392, 1394 (C.D. Cal. 1992).

130. *Sega*, 977 F.2d at 1522; see *supra* notes 89-97. The district court classified Accolade's use as purely commercial, which weighed in Sega's favor. *Sega*, 785 F. Supp. 1392, 1398 (N.D. Cal. 1992).

131. 17 U.S.C. § 107(4) (1988).

132. *Sega*, 977 F.2d at 1523.

133. The court in *Sega* acknowledged that, via reverse engineering, Accolade and other competitors increased the number of independently designed video games

create a virtual monopoly by eliminating legitimate competitors.¹³⁴ Such use of the Copyright Act flouts the constitutional purpose to advance technology by building on the ideas of others. Thus, the fourth factor favored Accolade.¹³⁵

The second fair use factor analyzes the nature of the copyrighted work.¹³⁶ As discussed above,¹³⁷ utilitarian and factual expression warrant weaker copyright protection than works of fiction.¹³⁸ Moreover, because the unprotected ideas are stored in unintelligible form and are not visible on the computer screen, the only means of access is via disassembly, which requires copying. "Because Sega's video game programs contain unprotected aspects that cannot be examined without copying, we afford them a lower degree of protection than more traditional literary works."¹³⁹ Thus, the second factor was held to favor Accolade.¹⁴⁰

Finally, the third statutory factor considers the amount of the copyrighted work that was copied.¹⁴¹ Here, Accolade copied the entire code, which weighs against it, but does not "preclude a finding of fair use."¹⁴² Therefore, Accolade's intermediate copies of Sega's code generated during disassembly were a fair use exception to copyright infringement action.

4. Atari v. Nintendo: *Reverse Engineering is Legal per the Federal Circuit*

Before the Ninth Circuit had published its opinion in *Sega*, the Federal Circuit Court of Appeals recognized that making intermediate copies is necessary to understanding the underlying

brought to market. *Id.* "It is precisely this growth in creative expression, based on the dissemination of other creative works and the unprotected ideas contained in those works, that the Copyright Act was intended to promote." *Id.*

134. *Id.* This appears to be an effective business strategy for Lotus. Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203 (D. Mass. 1992); Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990).

135. *Sega*, 977 F.2d at 1524.

136. 17 U.S.C. § 107(2) (1988).

137. See *supra* notes 104-08 and accompanying text.

138. *Sega*, 977 F.2d at 1524; see *supra* note 110 and accompanying text.

139. *Sega*, 977 F.2d at 1526.

140. *Id.*

141. 17 U.S.C. § 107(3) (1988).

142. *Sega*, 977 F.2d at 1526 (citing Sony Corp. of Am. Inc. v. Universal City Studios, Inc., 464 U.S. 417, 449-50 (1984) and Hustler Magazine, Inc. v. Moral Majority, Inc., 795 F.2d 1148, 1155 (9th Cir. 1986)).

ideas of a computer program, because the ideas are stored in incomprehensible form. Interpreting Ninth Circuit case law, the court in *Atari Games Corp. v. Nintendo of America, Inc.*, held that the copying entailed in reverse engineering could be a fair use.¹⁴³ The court stated that "an author cannot acquire patent-like protection by putting an idea, process, or method of operation in an unintelligible format and asserting copyright infringement against those who try to understand that idea, process, or method of operation."¹⁴⁴ The court drew additional support for allowing reverse engineering from the Supreme Court's recent decision in *Feist Publications, Inc. v. Rural Telephone Service Co., Inc.*, where the Court held that copyright does not prevent subsequent users from copying from a prior author's work those constituent elements that are not original "as long as such use does not unfairly appropriate the author's original contributions."¹⁴⁵

Although the Federal Circuit Court of Appeals held that reverse engineering was a fair use in some circumstances, the court did not find that Atari's conduct was worthy of the fair use defense. Atari had fraudulently obtained a copy of Nintendo's object code from the U.S. Copyright Office.¹⁴⁶ The fair use exception is an equitable doctrine that requires that the alleged infringer must obtain an authorized copy.¹⁴⁷ Because Atari sought this equitable exception with unclean hands, the court held that any copying was not a fair use.¹⁴⁸

After rejecting Atari's fair use defense, the court applied the Ninth Circuit's two-part analysis to determine if the programs

143. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 843-44 (Fed. Cir. 1992).

144. *Id.* at 842.

145. *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 350 (1991), *quoted in Atari*, 975 F.2d at 843.

146. *Atari*, 975 F.2d at 843-44. Atari had attempted to reverse engineer Nintendo's 10NES software security system that, like Sega's lockout code, prevented unlicensed games from being played on the Nintendo Entertainment System. *Id.* at 836. Atari first attempted to reverse engineer the 10NES code by electronically monitoring the ROM chip in which the code was stored, then by chemically peeling and reading the chip. After failing at both attempts, Atari fraudulently obtained a copy of the 10NES code from the U.S. Copyright Office by stating that they were engaged in present or prospective litigation with Nintendo. *Id.*

147. *Id.* at 843; *Harper & Row, Publishers, Inc. v. Nation Enter.*, 471 U.S. 540, 562-63 (1985).

148. *Atari*, 975 F.2d at 843.

were substantially similar.¹⁴⁹ The first part is an objective, extrinsic analysis, where the ideas of the two works are compared. If the ideas are substantially similar, the court analyzes expression using a subjective, intrinsic analysis.¹⁵⁰

Analytical dissection is used in both stages of the Ninth Circuit test. In the extrinsic phase, the court uses expert testimony and analytical dissection to identify copying by objectively comparing alleged similarities between the works.¹⁵¹ If similarity results from unprotected expression,¹⁵² the intrinsic substantial similarity test is not performed and the defendant is entitled to summary judgment as no infringement exists.¹⁵³

Under the intrinsic test, the trier of fact compares the protected core of expression with the allegedly infringing program to determine if the two works are substantially similar.¹⁵⁴ The subjective test is based on the testimony from an ordinary reasonable person who, for comparing computer programs, is a computer programmer.¹⁵⁵ Thus, expert testimony is used in both prongs.

Expert testimony showed substantial similarity between Atari's Rabbit program and Nintendo's 10NES program, far

149. *Id.* at 844-45.

150. *Id.* at 844; *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1020 (N.D. Cal. 1992).

151. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1020-21.

152. Judge Walker previously held that the scope of plaintiff's copyright is determined in the intrinsic phase, where the court must identify elements unprotected due to merger, *scenes a faire*, and other limiting doctrines. *Apple Computer, Inc. v. Microsoft Corp.*, 759 F. Supp. 1444, 1456 (N.D. Cal. 1991). However, he later implied that copyrightability must be determined before reaching the intrinsic test. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1020. Therefore, it appears that a court should first determine copyrightability, then perform the Ninth Circuit's objective and subjective substantial similarity tests, comparing the protected expression to the allegedly infringing work.

153. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1020; *accord* *Data East USA, Inc. v. Epyx, Inc.*, 862 F.2d 204, 209 (9th Cir. 1988). In a preliminary ruling in *Apple*, Judge Walker recognized that unprotected elements should be included in the intrinsic substantial similarity analysis, because excluding them eliminates copyright protection for a creative arrangement or compilation of unprotected expression. *Apple Computer, Inc. v. Microsoft Corp.*, 779 F. Supp. 133, 135 (N.D. Cal. 1991). Therefore, if the court finds no similarity between discrete elements, it does not even evaluate the program as a unique arrangement of unprotected elements or compilation. David A. Lowe, Comment, *A Square Peg in a Round Hole: The Proper Substantial Similarity Test for Nonliteral Aspects of Computer Programs*, 68 WASH. L. REV. 351, 363 (1993).

154. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1020-21.

155. *Atari*, 975 F.2d at 844.

beyond replication of the code necessary to unlock the Nintendo game console.¹⁵⁶ Thus, the court upheld the preliminary injunction against Atari because Nintendo was likely to be successful in its copyright infringement claims.¹⁵⁷

Atari and *Sega* were the first decisions to squarely deal with the legality of software reverse engineering. *Sega* provided a cogent framework for analyzing reverse engineering that realistically reflects commercial software practices, while protecting the copyright holder's rights against piracy. It is a critically important issue for the electronics industry because reverse engineering is a common industry practice.¹⁵⁸ Copyright protects an author's rights in his expression, but not access to his ideas.¹⁵⁹ The courts in *Atari* and *Sega* recognized that getting access to the ideas necessarily involves intermediate copying, which is legal when done on authorized copies merely to analyze the underlying ideas.¹⁶⁰ Thus, although *Sega* and *Atari* have

156. *Id.* at 845. The 10NES program contained several unused, prospective lockout codes which were also copied by Atari. *Id.* However, the unused codes were extraneous to Atari's stated fair use: reverse engineering to obtain the current lockout code. Incorporating these unnecessary instructions suggested illegal copying, not independent creation. *Id.* "The existence of the identical unnecessary instructions in both codes is strong proof of substantial similarity." *Id.* (quoting *E.F. Johnson Co. v. Uniden Corp. of Am.*, 623 F. Supp. 1485, 1496 (D. Minn. 1985)). Further, the court in *Atari* extended protection to the 10NES security system because an original program generated the "key." *Id.* The key was not purely functional because competitors could create their own "key" generators. *Atari*, 975 F.2d at 839. Prompted by *Sega's* suggestion that their TMSS should merit similar protection, the Ninth Circuit amended its opinion. *Id.* In a footnote, the court noted that *Sega's* console could be unlocked using only a single "functional" key, which did not merit copyright protection due to the functionality and "words and short phrases" doctrines. *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 n.7 (9th Cir. 1992).

157. *Atari*, 975 F.2d at 845.

158. *Sega Enter. Ltd. v. Accolade, Inc.*, 785 F. Supp. 1392, 1399 (N.D. Cal. 1992). Indeed, Congress expressly permitted reverse engineering for hardware (chip) development under the Semiconductor Chip Protection Act. 17 U.S.C. § 906 (1988).

159. *Atari*, 975 F.2d at 843-44. Under copyright law, ideas remain accessible to the public. Only through reverse engineering may the "public" access the ideas encrypted in object code. Gary R. Ignatin, Comment, *Let The Hackers Hack: Allowing The Reverse Engineering of Copyrighted Computer Programs to Achieve Compatibility*, 140 U. PA. L. REV. 1999, 2015 (1992).

160. *Atari*, 975 F.2d at 843. One commentator disputes that copying should be permitted to access a program's ideas, comparing object code to a foreign language book or a musical score which must be translated or performed to be comprehensible to most people. Miller, *supra* note 110, at 988-90. However, humans cannot remember and comprehend "the millions of zeros and ones that make up a program" without recording the translation. *Sega*, 977 F.2d at 1525. Such copying does violate the owner's copyright, but is excused as a fair use. Software innovators should have the same right to extract ideas as persons analyzing traditional copyrighted works.

narrowed the scope of copyright protection by permitting limited reverse engineering, these decisions will encourage technology development.

B. Protection of Nonliteral Expression: Structure, Sequence, and Organization

Copyright extends to some degree beyond the literal text to the work's nonliteral elements.¹⁶¹ "It is of course essential to any protection of literary property . . . that the right cannot be limited literally to the text, else a plagiarist would escape by immaterial variations."¹⁶² Most courts have protected the nonliteral elements of a program via two standards. First, courts have determined substantial similarity based on the structure, sequence, and organization of the two programs.¹⁶³ Second, the courts have focused on the user interface, how the program interacts with the user, or its "look and feel."¹⁶⁴

1. Structure, Sequence, and Organization as Protected Expression

In *Whelan Associates, Inc. v. Jaslow Dental Laboratories, Inc.*,¹⁶⁵ the Third Circuit Court of Appeals held that the nonliteral elements of a computer program, the structure, sequence, and organization, were copyright protectable by analogy to literary works, where nonliteral elements such as plot or plot devices are protected.¹⁶⁶ *Whelan* involved two computer programs that automated the business of operating a dental lab.

Ignatin, *supra* note 159, at 2009.

161. *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 701 (2d Cir. 1992); *Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1233 (3d Cir. 1986); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 46 (D. Mass. 1990).

162. *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931). Although *Nichols* was a purely literary case, Judge Learned Hand articulated basic principles for determining copyright infringement that modern courts have embraced in software infringement cases. *Id.* First, Judge Hand stated that copying of either literal or nonliteral elements of a work may constitute infringement. *Id.* Second, the appropriation must be "substantial." *Id.* Finally, Judge Hand developed his "abstractions" test for separating expression from the underlying ideas. *Id.* Abstraction is discussed further at *infra* notes 202-03.

163. *Whelan*, 797 F.2d at 1233.

164. *Paperback*, 740 F. Supp. at 37.

165. 797 F.2d 1222 (3d Cir. 1986).

166. *Id.* at 1234.

The two programs were written in different languages,¹⁶⁷ so no literal copying occurred, but the programs had similar structure in their file structures, screen outputs, and subroutines.¹⁶⁸

In its analysis, the court looked to *Baker v. Selden*, which first announced the idea-expression dichotomy.¹⁶⁹ In separating the idea from its expression, the court in *Baker* focused on the original work's purpose.¹⁷⁰ The test articulated by the court in *Whelan*, based on the *Baker* analysis, adopted the program's central function as its idea and regarded everything else as expression.¹⁷¹ The court bolstered its argument by holding that when there are multiple ways to achieve the same purpose or function, the means chosen is entirely expression.¹⁷² Because other, differently structured programs accomplished the same end as the defendant's, the court held that the structure did not merge with the idea. Therefore, it was protected expression.¹⁷³

In the second phase of analysis, the court held that the structure of the plaintiff's and defendant's programs had substantially similar file structures, subroutines, and screen outputs.¹⁷⁴

The court noted that in the delicate balance between competition and protection, the author's developmental efforts and costs should be rewarded by protecting his work.¹⁷⁵ By protecting nonliteral elements of the code, computer

167. *Id.* at 1226.

168. *Id.* at 1243-45.

169. 101 U.S. 99 (1879).

170. *Id.* at 101.

171. *Whelan*, 797 F.2d at 1236. The court in *Whelan* characterized the program's idea as "the efficient organization of a dental laboratory . . ." *Id.* at 1240. Under this broad definition, nearly any program that automated the basic business functions of running a dental laboratory would probably infringe on the plaintiff's work.

172. *Id.* at 1236.

173. *Id.* at 1238-39.

174. *Id.* at 1242-43. The court adopted a single substantial similarity test that admitted both expert and lay testimony, unlike traditional literary similarity tests based only on lay observations. *Id.* at 1233. Because computer programs are complex and usually foreign to the lay person, the ordinary observer test used to establish substantial similarity between literary works is not effective for computer programs. *Id.* at 1232-33; *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 844 (Fed. Cir. 1992).

175. *Whelan*, 797 F.2d at 1232. The assertion that effort should be rewarded with copyright protection, where the work would not otherwise qualify, has been expressly overruled by the Supreme Court in *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991).

programmers would have an incentive to create software.¹⁷⁶ This holding elevated the rights of authors with less regard to the legitimate scope of copyright protection afforded to the work itself. Thus, *Whelan* flipped the constitutional balance of granting authors limited rights in their work for the ultimate purpose of promoting science and the arts.¹⁷⁷

Whelan was a landmark case and many federal courts followed its reasoning. However, when courts began dealing with more technical copyright infringement issues, the weaknesses in *Whelan's* analysis became apparent.¹⁷⁸ First, *Whelan* advocated broad copyright protection to "provide the proper incentive for programmers by protecting their most valuable efforts"¹⁷⁹ However, the Supreme Court in *Feist* interpreted the Constitution as requiring that the author's rights must bow to the interests of progress and thus buried the "sweat of the brow" doctrine.¹⁸⁰

Second, *Whelan* assumed that each program had a single idea or purpose. Most software is very complex and performs several functions and many more subtasks as well. To characterize a program as having a single function is far too broad and grants protection to the many secondary ideas involved in the program.¹⁸¹

Finally, *Whelan* stated that the means used to convey the central idea was protectable expression, except expression necessary to implement the idea,¹⁸² or factual material that has a limited variation of expression.¹⁸³ "Where there are various means of achieving the desired purpose, then the particular means chosen is not necessary to the purpose; hence, there is

176. *Whelan*, 797 F.2d at 1237.

177. U.S. CONST. art. I, § 8, cl. 8.

178. Wald et al., *supra* note 36, at 5. Courts that have declined to follow *Whelan* include Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992); Sega Enter. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992); Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832 (Fed. Cir. 1992); Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc., 807 F.2d 1256 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987); CMAX/Cleveland, Inc. v. UCR, Inc., 804 F. Supp. 337 (M.D. Ga. 1992); Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1024 (N.D. Cal. 1992).

179. *Whelan*, 797 F.2d at 1237.

180. See *Computer Assoc.*, 982 F.2d at 711; *CMAX*, 804 F. Supp. at 352 (noting *Feist's* premise that primary objective of copyright is not to reward authors).

181. *Computer Assoc.*, 982 F.2d at 705.

182. *Whelan*, 797 F.2d at 1236.

183. *Id.* at 1237.

expression, not idea."¹⁸⁴ However, the analysis should not end here. For example, commands arranged in order of user frequency would be functional and not purely expressive.¹⁸⁵ A more practical rule was articulated in *Frybarger v. International Business Machines Corp.*: if the work is sufficiently functional such that the underlying idea is subject only to a limited range of expression, copyright protection extends only "against virtually identical copying."¹⁸⁶

The overbroad rule articulated in *Whelan* effectively grants copyright protection to unprotectable expression¹⁸⁷ and gives the author a patent-like monopoly without the rigors of the patent procedure.¹⁸⁸ This result is inconsistent with Section 102(b) of the 1976 Act which provides that copyright protection shall not extend to ideas, methods, or processes embodied in the work.¹⁸⁹

2. *Whelan Abandoned by Second Circuit*

Recognizing *Whelan's* flaws, the Second Circuit in *Computer Associates International, Inc. v. Altai, Inc.*,¹⁹⁰ wholly rejected *Whelan* and performed a careful and thoughtful analysis of the substantial similarity of two computer programs.¹⁹¹

Computer Associates (CA) developed and marketed a job scheduling program for IBM mainframe computers called CA-SCHEDULER.¹⁹² As part of the program, a subprogram called ADAPTER served as a translator between CA-SCHEDULER and

184. *Id.* at 1236.

185. Wald et al., *supra* note 36, at 25.

186. *Frybarger v. International Bus. Mach. Corp.*, 812 F.2d 525, 530 (9th Cir. 1987).

187. *Whelan* allows innovative programmers to monopolize efficient programming techniques, requiring only minimal creativity to achieve copyright protection. Menell, *supra* note 26, at 1082. As a result, subsequent programmers must risk infringement by employing these efficient techniques and subsequent program efficiency declines. *Id.*; see *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 345 (1991); *Atari Games Corp. v. Oman*, 979 F.2d 242, 242-43 (1992) ("[T]he requisite level of creativity [for copyrightability] is extremely low."). This is hardly the result intended by the Constitution's framers; see also *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 705 (2d Cir. 1992); *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524-25 (9th Cir. 1992).

188. Wald et al., *supra* note 34, at 5.

189. *Id.*; 17 U.S.C. § 102(b) (1988).

190. 982 F.2d 693 (2d Cir. 1992).

191. Cf. Miller, *supra* note 110, at 1001-02 (characterizing *Computer Assoc.* as a refinement of, rather than a departure from, *Whelan*).

192. *Computer Assoc.*, 982 F.2d at 698.

the host computer's operating system.¹⁹³ Because of the translator subprogram, the user could utilize the same software on several different operating systems, eliminating the need for multiple programs for each operating system. CA-SCHEDULER was very popular.¹⁹⁴

In 1982, Altai began marketing ZEKE, a job scheduling program for the VSE operating system. In response to consumer demand, Altai decided to create a version of ZEKE compatible to the MVS operating system.¹⁹⁵ Altai recruited a CA designer, Claude Arney, to work on the new program. In violation of the trade secret agreements he executed with CA, and unknown to Altai, Arney brought with him the source code for ADAPTER, portions of which he integrated into the new Altai program OSCAR 3.4.¹⁹⁶

CA learned of Altai's infringement, obtained copyrights on its own products, and sued Altai and Arney for copyright infringement. When Altai learned of the suit, it rewrote OSCAR 3.4 without Arney or the ADAPTER code, and shipped the new product, OSCAR 3.5, to customers as a free upgrade.¹⁹⁷

CA won the infringement suit on OSCAR 3.4, but lost on OSCAR 3.5 and appealed to the Second Circuit.¹⁹⁸ CA asserted that the district court had erroneously applied a substantial similarity test that did not sufficiently account for CA's nonliteral program elements, particularly the structure of the two programs.¹⁹⁹

The Second Circuit in *Computer Associates* criticized the *Whelan* analysis as impractical and overbroad²⁰⁰ and advocated a three-part test to determine nonliteral substantial similarity: (1) abstraction, (2) filtration, and (3) comparison.²⁰¹ In the abstraction step, the court first dissects the program into distinct functional levels. At each level, the court then characterizes each element by a series of abstract descriptions, moving from the code itself (specific) to the program's ultimate function

193. *Id.*; see *supra* note 40.

194. *Computer Assoc.*, 982 F.2d at 699.

195. *Id.*; see *supra* note 40.

196. *Computer Assoc.*, 982 F.2d at 699-700.

197. *Id.* at 700.

198. *Id.*

199. *Id.* at 701.

200. *Id.*

201. *Id.* at 706-11.

(general).²⁰² From these abstractions, the court determines where the basic ideas end and where the creative expression begins.²⁰³

The second step is successive filtration. Here, the court examines the structure at each level of abstraction for unprotectable material, thus leaving a core of protected expression.²⁰⁴ This step defines the scope of the copyright.²⁰⁵ The structure at each level of abstraction is examined for ideas, expression necessarily incidental to the idea, expression controlled by external factors and desire for efficiency, and expression in the public domain. This enumerated material is filtered out as unprotected by copyright.²⁰⁶

Finally, the core of protectable expression is then compared with the allegedly infringing program to determine if substantial similarity exists between them. If the similarities stem from common ideas and not common expression, then no infringement has occurred.²⁰⁷

The court in *Computer Associates* also examined the delicate balance between "competition and protection."²⁰⁸ Although *Whelan* purported to motivate programmers by protecting their developmental efforts, the Supreme Court held five years after *Whelan* that effort alone does not confer copyright protection, thus rejecting the "sweat of the brow" protection.²⁰⁹ The court in *Computer Associates* noted that the "primary objective of copyright is not to reward the labor of authors," but to promote the sharing of ideas to advance progress and science for the public good.²¹⁰ Thus, the intensive labor involved in designing flow charts, creating macros and parameter lists, and debugging does not by itself confer copyright protection.²¹¹

202. *Id.* at 706-07.

203. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 839 (Fed. Cir. 1992).

204. *Computer Assoc.*, 982 F.2d at 707 (quoting 3 NIMMER ON COPYRIGHT, § 13.03[F][5], at 13-78.44 (1992)).

205. *Id.* at 707.

206. *Id.*; see *supra* notes 33-43 and accompanying text.

207. *Computer Assoc.*, 982 F.2d at 711 (quoting *Data East USA, Inc. v. Epyx, Inc.*, 862 F.2d 204, 208 (9th Cir. 1987)).

208. *Id.*

209. *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991).

210. *Id.* at 1290, quoted in *Computer Assoc.*, 982 F.2d at 711.

211. *Computer Assoc.*, 982 F.2d at 711.

In addition, the court in *Computer Associates* blessed the use of expert testimony to assist the trier of fact to determine substantial similarity. While expert testimony had previously been allowed to ascertain whether the defendant had copied the plaintiff's work, prior courts admitted lay testimony to determine whether or not the copying was unlawful.²¹² Because computer programs are highly technical and complex, the court held that expert testimony may be used to determine substantial similarity in software copyright infringement cases at the discretion of the district courts.²¹³ To this limited extent, *Whelan* and *Computer Associates* are in harmony.

Computer Associates has already affected software copyright litigation, receiving mostly favorable comment.²¹⁴ It is not the death knell for SSO litigation, but the scope of protection will narrow, appropriately, to cover only those elements that are truly protectable.²¹⁵ This analysis comports with the constitutional intent to promote science and progress with the free flow of ideas.

C. Protection of Nonliteral Expression: A Program's "Look and Feel"

Another area that has been a source of copyright confusion is a program's "look and feel," or how the program interfaces with the user. A program's user interface is often "of greater commercial value" than the underlying code.²¹⁶ In fact, the user interface is

212. *Id.* at 713.

213. *Id.*

214. The Second Circuit acknowledged many programming realities, such as program structure and sequence, are often dictated by efficiency and external compatibility requirements. *Id.* at 707-10. "Efficient" expression merely performs a function that may only be expressed in a very few technologically acceptable ways. *Computer Assoc.*, 982 F.2d at 708. Where efficiency dictates expression, the function or idea has merged with its expression, and the doctrine of merger precludes copyright protection. *Id.* By refusing to protect such nonliteral components, the court promoted programming efficiency because programmers can spend time innovating, rather than "reinventing the wheel." Dennis M. McCarthy, *Recent Decision*, 66 TEMP. L. REV. 273, 285 (1993); Ignatin, *supra* note 159, at 2030; cf. Miller, *supra* note 110, at 1004 (criticizing *Computer Assoc.*'s filtering out of efficient expression); see also *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1525 (9th Cir. 1992); *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 839 (Fed. Cir. 1992); *CMAX/Cleveland, Inc. v. UCR, Inc.*, 804 F. Supp. 337 (M.D. Ga. 1992); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992); *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1025 (N.D. Cal. 1992).

215. *Computer Assoc.*, 982 F.2d at 711.

216. *Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1231 (3d Cir.

often developed first, then the code is written to implement it.²¹⁷

Screen displays may be copyrighted as audiovisual works,²¹⁸ a separate entity from the program itself, which is copyrighted as a literary work.²¹⁹ However, courts have protected interface details such as command names, structure and hierarchy, choice of keystrokes, and menu organization as nonliteral elements of the copyrighted program, apart from the visual display itself.²²⁰ The analysis is similar to that used in the SSO cases above, but the programs are examined on their external structure or appearance to the user, rather than on their internal code structure.

1. *Lotus Cases Expand Copyright Protection for "Look and Feel"*

In *Lotus Development Corp. v. Paperback Software International*,²²¹ the District Court of Massachusetts held that Paperback had infringed on Lotus 1-2-3's copyrighted user

1986). Competitors, seeking to emulate successful software programs, often start by creating a similar "look & feel," then work backwards to develop the necessary flowcharts and source code. *Id.* To woo buyers away from the original product, competitors incorporate innovations, thus furthering progress. *Id.*

217. Levi, *supra* note 116, at 2.

218. 17 U.S.C. § 102(a)(6) (1988).

"Audiovisual works" are works that consist of a series of related images which are intrinsically intended to be shown by the use of machines, or devices such as projectors, viewers, or electronic equipment, together with accompanying sounds, if any, regardless of the nature of the material objects, such as films or tapes, in which the works are embodied.

Id. § 101 (1988).

219. *Id.* § 102 (1988). In 1987, the Copyright Office adopted a single registration policy to cover both screen displays and the underlying code. Registration and Deposit of Computer Screen Displays, 53 Fed. Reg. 21,817, 21,818 (1988) (codified at 37 C.F.R. § 202); Levi, *supra* note 116, at 4. The Office supported single registration because of the potential confusion and multiple claims that may arise from "piecemeal" registrations. *Id.*; Matthew J. Fortnow, Note, *Why The "Look and Feel" of Computer Software Should Not Receive Copyright Protection*, 14 CARDOZO L. REV. 421, 423-33 (1992).

220. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. 78 (D. Mass. 1992); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990); *Digital Communications Assoc., Inc. v. Softklone Distrib. Corp.*, 659 F. Supp. 449 (N.D. Ga. 1987). The Copyright Act defines a computer program as a "set of statements or instructions . . ." 17 U.S.C. § 101 (1988). Some argue that "look and feel" should not be protected because it does not fall within the letter of the Act. Fortnow, *supra* note 219, at 421.

221. 740 F. Supp. 37 (D. Mass. 1990).

interface, composed of the menu structure and organization, long prompts, function key assignments and macro commands, all nonliteral elements of the program.²²² Paperback, attempting to tap into Lotus' commercial success, independently designed its electronic spreadsheet program to "look and feel" like Lotus 1-2-3. Paperback did not attempt to copy or reverse engineer the literal elements of Lotus 1-2-3, but essentially copied the user-manipulated interface.²²³

The court in *Paperback* advocated a three-part analysis to determine copyrightability.²²⁴ The *Paperback* analysis is similar to the abstraction and filtration steps of the *Computer Associates* analysis.²²⁵ However, in *Computer Associates* abstraction phase, the court first dissected the program into discrete components, then extracted the expression.²²⁶ In *Paperback*, Judge Keeton emphatically stated that the copyrighted work should not be dissected. "Rather, the court need only identify those elements that are copyrightable, and then determine whether those elements, *considered as a whole*, have been impermissibly copied."²²⁷ Under the *Paperback* test, copyright protection extends to a creative selection or arrangement of otherwise unprotected expressive elements.²²⁸ Therefore, the user

222. *Id.* at 63.

223. *Id.* at 69.

224. *Id.* at 60-61. In the first step, the court uses abstraction to separate expression from the underlying ideas, systems, or procedures. *Id.* Second, the court must identify protected expression using the doctrines of merger and indispensable expression. *Id.* Finally, the court compares the core of protected expression to the copyrighted work as a whole. *Id.* This step acknowledges that a quantitatively small amount of protected expression may be the most qualitatively valuable part of the work. *Id.* Judge Keeton further explains and defends this test in the *Borland* decisions. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 202 (D. Mass. 1993); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203, 216-19 (D. Mass. 1992).

225. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. at 211. *See supra* notes 203-06 and accompanying text.

226. *Id.*

227. *Paperback*, 740 F. Supp. at 67 (citing with approval *Atari Games Corp. v. Oman*, 888 F.2d 878, 882-83 (D.C. Cir. 1989)). In contrast, the *Computer Assoc.* test compares the kernel of protected expression to the allegedly infringing work to determine substantial similarity. *Id.* This method ignores the copyrightability of a creative arrangement of unprotected elements. *Id.* However, the court in *Computer Assoc.* noted that its decision did not "control infringement actions regarding categorically distinct works, such as certain types of screen displays." *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 703 (2d Cir. 1992); *see* David L. Hayes, *What's Left of "Look and Feel": A Current Analysis* (pt. 2), *COMPUTER LAW.*, June 1993, at 1, 5.

228. This notion reflects the copyrightability of compilations permitted in the

interfaces and menu command structure, or the "look and feel" of a program, would be protected as unique expression, even if the individual elements would not.²²⁹

The court rejected Paperback's functionality defense, stating that copyright protection does not cease when the copyrighted expression becomes the market standard of utility.²³⁰ To do so, the court held, would extend protection only to second-rate expression.²³¹ Further, even if some of the elements, such as commands, are quite obvious and functional, a court should consider the interface as a single entity to determine if it is wholly incidental to the ideas conveyed (unprotected) or if it is a creative expression as a whole (protected).²³²

After its success against Paperback, Lotus turned its fury on another competitor, Borland, and again proved that a competitor had infringed on Lotus 1-2-3's user interface.²³³ Borland, like Paperback, had independently created an electronic spreadsheet program to "look and feel" like Lotus 1-2-3. In *Lotus Development Corp. v. Borland International, Inc.*, Judge Keeton announced a three-part test to determine copyright infringement.²³⁴ To prove infringement, the plaintiff must show: (1) evidence of copying,²³⁵ (2) the scope of copyright protection (copyrightability),²³⁶ and (3) substantial similarity between the protected expression and the allegedly infringing work (illicit copying).²³⁷

Unlike in *Paperback*, Lotus did not prove that Borland copied its entire interface,²³⁸ but contended that the menu commands

Copyright Act. 17 U.S.C. § 101 (1988). "A 'compilation' is a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship." *Id.*; see *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 356-57 (1991); see also Miller, *supra* note 110, at 1003.

229. *Paperback*, 740 F. Supp. at 67.

230. *Id.* at 58. But see *infra* notes 259-60 and accompanying text.

231. *Paperback*, 740 F. Supp. at 58.

232. *Id.* at 67.

233. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992).

234. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. 78, 81, 87 (D. Mass. 1992). In *Paperback*, the defendants admitted copying, *Paperback*, 740 F. Supp. at 68, so to establish infringement, Lotus had to prove only that their user interface was copyrightable as a whole. *Id.* at 70; *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. at 81.

235. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. at 208.

236. *Id.* at 209.

237. *Id.* at 220.

238. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. at 81.

and structure were copied.²³⁹ The court agreed that Borland had copied protected elements of the 1-2-3 interface.²⁴⁰ Thus, Borland infringed as a matter of law.²⁴¹ At trial to determine the scope of illicit copying, the court concluded that Borland's menu trees were substantially similar to the 1-2-3 menu tree.²⁴²

Although Lotus' claims in *Borland* forced a more elemental infringement analysis than was required in *Paperback*, Judge Keeton rejected Borland's arguments that functionality, compatibility,²⁴³ and standardization requirements constrained their choice of menu commands and structure.²⁴⁴ The court also rejected Borland's fair use defense, ruling that its use was essentially commercial and competitive.²⁴⁵ Therefore, copyright protection shows no signs of weakening in Judge Keeton's court.

239. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 202, 209 (D. Mass. 1993).

240. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. at 205, 223.

241. *Id.* at 223.

242. Judge Keeton analyzed substantial similarity, or the scope of infringement, based on the degree of copying and "the nature of the copied work." *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. at 209. He determined that neither functional considerations, *id.* at 215, nor lack of originality limited Borland's choice of menu commands and structure. *Id.* at 217. In copyright terms, "originality" means "that the work was independently created by the author" and "that it possess at least some minimal degree of creativity." *Id.* at 215. The requisite spark of creativity to justify copyright protection is small. *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 345 (1991); *Atari Games Corp. v. Oman*, 979 F.2d 242, 242-43 (D.C. Cir. 1992). The court also held that Borland's Key Reader feature infringed because it contained the 1-2-3 menu in its code, even though it was invisible to the user. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. at 230.

243. Judge Keeton did acknowledge that compatibility may be a limiting externality if the original program pre-existed the copyrighted work. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. at 213. See Hayes, *supra* notes 227, at 4.

244. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. at 216-17; *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. at 231-32. Although Judge Keeton suggested alternative menu terms such as "Ditto," "Clone," or "Duplicate" instead of Lotus' choice, "Copy," he ignored software reality. Terms such as "Copy," "Move," "Print," and "File" are software terms in the public domain. "Plaintiff's [sic] may not claim copyright protection of an idea and expression that is, if not standard, then commonplace in the industry." D. Lee Anton & Brett N. Dorny, *How Copyrightability Trivialized Substantial Similarity: The Significance of Lotus v. Borland*, *COMPUTER LAW.*, Oct. 1992, at 6, 10 (quoting *Brown Bag Software v. Symantec Corp.*, 12 U.S.P.Q.2d 1991, 1995 (N.D. Cal. 1989), *aff'd*, 960 F.2d 1465 (9th Cir. 1992)). Users are accustomed to these commands in a variety of programs. Judge Keeton's suggestions are as awkward as encouraging car manufacturers to adopt new commands such as "Back Up" for "Reverse," "Stand" for "Park," "Disengage" for "Neutral," and "Cruise" for "Drive."

245. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 223, 245 (D. Mass. 1993).

2. *The Ninth Circuit Limits "Look and Feel" Protection*

In stark contrast to the *Paperback* and *Borland* decisions and only one week after *Borland*, the District Court for the Northern District of California granted summary judgments to Microsoft and Hewlett-Packard on actions for copyright infringement of Apple's graphical user interface.²⁴⁶ Apple had asserted that Microsoft's Windows program infringed on the "look and feel" of Apple's Macintosh by using similar symbolic icons, pull-down menus, icon names, and text directories, and use of nested windows, which the user could move and resize.²⁴⁷

Apple had settled a previous, similar suit with Microsoft on the first Windows program, Windows 1.0.²⁴⁸ As a result, Apple granted a non-exclusive license to Microsoft for the Windows 1.0 displays. Later, Apple alleged additional infringement of their audiovisual copyrights based on Microsoft's Windows 2.03 and 3.0 and Hewlett-Packard's NewWave 3.0.²⁴⁹

The court employed the Ninth Circuit's two-part test to determine substantial similarity.²⁵⁰ The court held that Apple and Microsoft's previous agreement covered many similarities.²⁵¹ The court deemed most of the remaining expression unprotected due to the merger, *scenes a faire*, and limited ways of expression doctrines.²⁵² The remaining similarities not knocked out by the limiting doctrines above were then analyzed intrinsically.²⁵³

In contrast to the court in *Borland*, the court in *Apple* recognized that several expressions were not protected due to

246. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1041-42 (N.D. Cal. 1992). The court granted Microsoft's summary judgment motions on a majority of elements after determining that they were unprotected, using analytical dissection in the extrinsic analysis. *Id.* Of over 150 alleged similarities, only nine remained in dispute after the court's Aug. 7, 1992 decision. *Apple Computer, Inc. v. Microsoft Corp.*, 821 F. Supp. 616, 619 (N.D. Cal. 1993). Performing the intrinsic analysis, the court analyzed five elements using a "virtually identical" standard and four elements using the "substantial similarity" standard. *Id.* Apple's user interface was copyrighted as an audiovisual work, not as a literary work. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1015.

247. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1017.

248. *Id.* at 1015.

249. *Id.* at 1015-16.

250. *Id.* at 1020; see *supra* notes 150-55 and accompanying text.

251. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1016.

252. *Id.* at 1041-42; see *supra* notes 33-43 and accompanying text.

253. *Apple Computer, Inc. v. Microsoft Corp.*, 821 F. Supp. 616, 619 (N.D. Cal. 1993) (performing intrinsic analysis).

their utilitarian nature. Because there are limited ways to express a function efficiently, "copyright law will abhor only a virtual copy of the original."²⁵⁴ Thus, the district court applied two standards of infringement in its intrinsic analysis: a "substantial similarity" test for protectable expression²⁵⁵ and a "virtually identical"²⁵⁶ test for expression that is otherwise unprotected due to copyright limiting doctrines.²⁵⁷

In employing dissection in the extrinsic analysis, the court in *Apple* did not adopt the holistic comparison approach used in both *Lotus* cases. The *Apple* decision held that the arrangement of functional elements, or "look and feel," was itself functional and unprotected as a whole.²⁵⁸ The court held that the similarity in arrangement of functional elements does not signify copying, but standardization. If a display becomes so associated with its functional purpose that it becomes a market standard, it may be termed an idea.²⁵⁹ This is in direct conflict with the reasoning of the *Lotus* cases.²⁶⁰

The dissection analysis used by the court in *Apple* is fully endorsed by the Second Circuit's analysis in *Computer Associates* and the Ninth Circuit's analysis in *Brown Bag Software v. Symantec Corp.*²⁶¹ In *Brown Bag*, the Ninth Circuit made clear

254. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1021.

255. *Apple Computer, Inc. v. Microsoft Corp.*, 821 F. Supp. at 623.

256. *Id.* at 619.

257. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1021-22; *Apple Computer, Inc. v. Microsoft Corp.*, 821 F. Supp. at 623.

258. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1023; cf. Miller, *supra* note 110, at 988. Architecture, like computer programs, combines aesthetics and function to achieve utilitarian ends. Architecture is similarly protected by copyright. *Id.*

259. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. at 1023; see *supra* notes 33-42 and accompanying text. However, in rejecting Apple's "look and feel" argument, Judge Walker failed to consider Apple's arrangement of unprotected elements as a compilation. Bradford P. Lyerla, *Copyrightability of Software User Interfaces: The Natural Law Versus the Social Utilitarian Approach*, COMPUTER LAW., Jan. 1993, at 21, 23.

260. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990). One commentator has characterized Judge Keeton (*Lotus* cases) as an advocate of "natural law" and Judge Walker (*Apple* cases) as a "social utilitarian." Lyerla, *supra* note 259, at 23. The *Lotus* decisions represent a rather Darwinian "survival of the fittest" (or at least "survival of the first to market") theory, regardless of the underlying purpose of copyright law. *Id.* At the other extreme, Walker appears to minimize the copyright holder's incentives and property rights in order to encourage incremental innovation and competition. *Id.*

261. 960 F.2d 1465 (9th Cir. 1992). *Brown Bag* alleged that Symantec's "Grandview"

its intention to "perpetuate 'analytical dissection' as a tool for comparing not only ideas but also expression."²⁶² In performing its two-step analysis,²⁶³ the Ninth Circuit held that the extrinsic test involved dissecting the copyrighted work, separating idea from expression, distinguishing protected from unprotected expression, and then comparing the similar ideas and expressions between the two works.²⁶⁴ The scope of copyright is determined by distinguishing protected expression from unprotected ideas and expression,²⁶⁵ much like the abstraction and filtration steps of the *Computer Associates* analysis.²⁶⁶ Infringement results only if the similarities between the two programs result from protected expression.²⁶⁷

In *Brown Bag*, the court held that no infringement existed because any similarities arose from unprotected expression.²⁶⁸ The court declined to credit *Brown Bag*'s contention that the lower court failed to consider infringement of their program's "look and feel." The Ninth Circuit merely stated that *Brown Bag* had not requested that analysis from the district court, so it need not be performed.²⁶⁹ However, the court did concede in a footnote that "copyright may inhere, under appropriate circumstances, in the selection and arrangement of unprotected components."²⁷⁰ Thus, the *Brown Bag* decision limits, but does not eliminate, a "look and feel" analysis, unlike the wholesale rejection of "look and feel" in *Apple*.

infringed *Brown Bag*'s computer outlining program, "PC-Outline." *Id.* The Ninth Circuit affirmed the district court's holding that the similarities between the two programs resulted from unprotected ideas inherent to an outlining program. *Id.* at 1476.

262. *Id.* at 1475.

263. See *supra* notes 150-55 and accompanying text.

264. *Brown Bag*, 960 F.2d at 1475.

265. *Id.* at 1475-76.

266. See *supra* notes 201-06 and accompanying text.

267. *Brown Bag*, 690 F.2d at 1476; see *Data East USA, Inc. v. Epyx, Inc.*, 862 F.2d 204, 209 (9th Cir. 1988) (holding that no infringement exists when similarities result from unprotected expression).

268. *Brown Bag*, 690 F.2d at 1476.

269. *Id.*

270. *Id.* at 1476 n.4 (quoting NIMMER ON COPYRIGHT § 13.03[F][5], at 13-78.44 n.342). The D.C. Circuit Court of Appeals recently endorsed the copyrightability as an audiovisual work of a video game's "look and feel." *Atari Games Corp. v. Oman*, 979 F.2d 242 (D.C. Cir. 1992). Although the individual elements of Atari's display screens were uncopyrightable, *id.* at 244, compilations and audiovisual works were protected as an arrangement of related elements. *Id.* at 244-45.

IV. RECOMMENDATIONS AND CONCLUSIONS

A. *Program Standardization and Compatibility*

Due to the inevitable user gravitation to a popular program, de facto standards will emerge, regardless of the scope of copyright protection.²⁷¹ Consumer demands for compatibility and interoperability increase external factors, which decrease a programmer's options. Some commentators argue that copyright should not be used as a weapon to inhibit inevitable standardization and choke off incremental innovation.²⁷² Copyright law protects a program's literary content, not its utilitarian functions.²⁷³ To balance the competing interests of rewarding innovators and furthering technology, copyright law should permit only copying necessary to achieve

271. Unpopular programs tend to survive only one or two generations, leaving users stranded. Ignatin, *supra* note 159, at 2025, 2033. Users are sufficiently fearful of being burned that they flock to more popular programs. *Id.* As a program attracts more users, it becomes a "de facto standard," and the copyright owner enjoys a relative monopoly in the market. *Id.* To survive, competitors must offer programs that are compatible with the user's skills, expectations, and existing data files. *Id.* The desire to provide compatible products ignited the litigation in each case discussed in this Note. Timothy S. Teter, Note, *Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases*, 45 STAN. L. REV. 1061, 1061 (1993).

272. Menell, *supra* note 26, at 1053, 1066. Compatibility encourages innovation in incremental steps (evolution) rather than in technological leaps (revolution). *Id.* Patent better protects leaps. Ignatin, *supra* note 159, at 2030. Standardization of computer programs is important to both software developers and users. *Id.* The user's cost of changing software packages goes far beyond purchase price. *Id.* When software consumers factor in the additional costs of installation, training, file conversion, and support, they may be reluctant to change. *Id.* at 2028-29. To overcome this inertia, developers attempt to create "user-friendly" programs that capitalize on the user's prior knowledge and skill. *Id.* Thus, "user-friendliness" is enhanced by standardization, benefitting both user and developer. Menell, *supra* note 26, at 1066-67. Other commentators argue that only truly functional or merged expression is not copyrightable. *Id.* Therefore, although protected expression becomes "standard," it still merits protection. Miller, *supra* note 110, at 990. For example, no court would invalidate a patent simply because an invention became "popular."

273. Menell, *supra* note 26, at 1072; Ignatin, *supra* note 159, at 2021. Programs are not written to stimulate thoughtful discourse, create imagery, or evoke emotions as are traditional literary works, but to control the computer and provide the interface between the hardware and the human operator. *Id.* Every line of code is utilitarian and crafted for efficiency. *Id.* As seen in *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203, 218 (D. Mass. 1992), overbroad protection also protects user-interface arrangements deemed efficient by human factors analysis. Menell, *supra* note 26, at 1053. To limit free-riders in these cases, copyright law should protect against nearly literal copying. *Id.* at 1082. Further protection encourages monopolistic pricing and hobbles technology growth. *Id.* at 1080.

compatibility.²⁷⁴ Where externalities such as hardware and software compatibility, industry standards, and efficient programming methods leave programmers but a few ways to express a function, only virtually identical copying should be deemed infringement.²⁷⁵

Although in some jurisdictions a copyright owner's property interest appears to diminish once the program achieves "industry standard" status,²⁷⁶ the programmer is still motivated to develop new technology rather than steal existing expression.²⁷⁷ Acknowledging standardization does not license piracy.²⁷⁸ However, a competitor who attempts to improve on the "industry standard" may infringe by creating an unauthorized derivative.²⁷⁹

274. Teter, *supra* note 271, at 1068.

275. *Id.* The *Lotus* cases rejected compatibility as a justification for copying while the courts in *Apple* and *Sega* bought it. *Id.* Although some commentators argue that copyright law should provide the same protection to all works, Miller, *supra* note 110, at 1022, the Supreme Court has recognized that degrees of copyright protection exist. *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349 (1991) (explaining that copyright protection of a compilation may be "thin"); *see Atari Games Corp. v. Oman*, 979 F.2d 242, 244 n.4 (D.C. Cir. 1992) (discussing relative strengths of protection).

276. Existing copyright law does not provide for this change in status. Teter, *supra* note 267, at 1077. To permit such copying, the expression must have merged with the idea or be considered a function, procedure, or process. *Id.* However, the court must adopt the theory of dynamic merger, where the idea and expression merge over time. *Id.* Merger is analyzed at the time of infringement, not at inception. *Id.* at 1088.

277. An innovator enjoys the profits reaped during a significant leadtime in the market. Menell, *supra* note 26, at 1081 n.213. That leadtime may be the time until the program becomes the de facto standard or the time required to reverse engineer and emulate it. *Id.* Some commentators call for little protection beyond literal copying, stating that the originator's leadtime in the market, short market lifespan, and relatively low development costs militate against extensive protection. *Id.* Others argue that traditional copyright protection should not be compromised by standardization. Miller, *supra* note 110, at 990. Even when the program becomes "standard," it may be protected against identical copying as a compilation. Christian H. Nadan & James W. Morando, *Standardization and Interoperability Become Key Factors in Copyright Law*, COMPUTER LAW., Apr. 1993, at 12, 16.

278. Ignatin, *supra* note 159, at 2036. 17 U.S.C. § 106 (1988).

279. Among the copyright holder's exclusive rights is the right to create derivative works. 17 U.S.C. § 106 (1988). Programs that only subtly improve an existing program may be labelled derivative works. *Id.* Permitting standardization only shifts litigation to the issue of derivative works.

B. Proposed Test for Copyrightability and Substantial Similarity

The circuits desperately need to adopt standard tests for copyrightability and substantial similarity as well as guidelines for protecting nonliteral expression.²⁸⁰ The Second Circuit's three-step abstraction-filtration-comparison test has received very favorable reviews.²⁸¹ Yet, this test is not sufficiently comprehensive to protect all program expression.²⁸² As noted in *Computer Associates*, a court must perform an abstraction analysis at every structural level.²⁸³ As the elements under analysis become increasingly smaller, their expression merges with idea, or the ways of expressing the discrete idea become very limited. At minute levels, very little protectable expression remains.²⁸⁴ Thus, unless a court performs abstraction at various micro to macro levels, the program can be reduced to a collection of unprotected elements.

Opponents of analytical dissection fear that judges examining the trees lose sight of the forest.²⁸⁵ However, by performing an abstraction analysis at every program level, including an analysis of the overall arrangement, the court should see both the trees and the forest.

The test proposed in this Note is a two-part test, analyzing first copyrightability, then substantial similarity employing standard doctrines used to determine infringement in literary cases.²⁸⁶ In a copyrightability analysis, the court should first perform an abstraction test at each structural level of the program to extract expression from the underlying ideas,

280. See *supra* notes 150-55, 268-70 and accompanying text.

281. See *supra* note 214.

282. For example, critics of the *Computer Assoc.* test argue that it prematurely filters out unprotected expression, thus precluding compilation analysis. Lowe, *supra* note 153, at 364-67. Further, levels of abstraction are not clearly defined, leaving courts wide latitude to implement the test. *Id.*

283. *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 707 (2d Cir. 1992). Analysis at the highest levels of abstraction, including the work as a whole, primarily analyzes SSO and "look and feel," where the whole may be greater than the sum of its parts. *Id.*

284. Crowe, *supra* note 18, at 226; Lowe, *supra* note 153, at 368 (warning that the court may break down the program into such small increments that limiting doctrines eliminate copyrightability). Whelan illustrates the opposite problem. *Id.* There, the court analyzed the program at issue at only the highest level of abstraction, holding that virtually the entire program was a protectable expression of a single idea. *Id.*

285. See Lowe, *supra* note 153, at 364-67.

286. See also *id.* at 368-71 (advocating two-part substantial similarity test for nonliteral elements: (1) program-as-a-whole, and (2) elemental analysis).

methods, or processes. The resulting strata of expression should range from the totality of the program to its most discrete elements.

Filtration is the second step of the copyrightability test. Here, the court should determine the scope of copyright protection at each level by applying limiting doctrines such as merger, *scenes a faire*, expression in the public domain, limited ways of expression, functionality, and external constraints.²⁸⁷ After filtration, the court is left with a skeleton of protected expression. At the lowest levels of abstraction, only protected elements survive filtration. At the highest levels of abstraction, some individual elements may be unprotected, but they may be uniquely arranged in a copyrightable manner. Thus, this test permits copyrightability of both expressive components and their original creative arrangement or compilation.

Once the scope of copyright has been determined, the court must compare the plaintiff's protected expression to the defendant's allegedly infringing work. The strength of protection should depend on the copyrightability of both discrete elements of expression and their arrangement. Where protected elements or their unique arrangement has been copied, the court should employ the lesser test of substantial similarity. Where an original compilation of unprotected elements has been copied, only virtually identical copying should constitute infringement.²⁸⁸ Where similarities result from unprotected expression and not an original arrangement, there is no infringement.²⁸⁹

Due to the complexity of the material, courts should admit expert testimony at all levels of analysis.²⁹⁰ Using lay

287. Externalities that should be considered in the filtration step include hardware compatibility, user compatibility (standardization), efficiency, and standard programming practices. Hayes, *supra* note 227, at 11. SSO protection is likely to suffer most when externalities are filtered out, as much of a program's SSO may be dictated by these constraints.

288. Apple Computer, Inc. v. Microsoft Corp., 821 F. Supp. 616 (N.D. Cal. 1993); Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006 (N.D. Cal. 1992) (employing both substantial similarity and virtually identical standards); *accord* Atari Games Corp. v. Oman, 979 F.2d 242 (D.C. Cir. 1992); Frybarger v. International Bus. Mach. Corp., 812 F.2d 525, 530 (9th Cir. 1987) (advocating virtually identical standard for copying of unprotected elements).

289. Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 711 (2d Cir. 1992); Data East USA, Inc. v. Epyx, Inc., 862 F.2d 204, 208 (9th Cir. 1987).

290. See *supra* note 64.

testimony to determine similarities between books does not work when comparing highly technical, complex computer code.²⁹¹

Because identical screen displays can be generated by vastly different source codes, the displays and codes should be analyzed separately for copyrightability and substantial similarity.²⁹² Despite the Copyright Office's mandate for single registrations,²⁹³ the court in *Manufacturers Technologies, Inc. v. CAMS, Inc.*²⁹⁴ recognized that

a computer program and its screen displays are, for copyright purposes, fundamentally distinct. The computer program and any authorship contained therein is designed to organize and direct the computer to efficiently perform a particular task when properly directed by the user. While the user interface is designed to communicate with the user in a way to facilitate the understanding and use of the program itself [sic].²⁹⁵

The holistic "look and feel" analysis should be abandoned,²⁹⁶ because it too readily protects expression that is unprotectable, and does not allow for subtle improvements in expression. Plaintiffs seeking "look and feel" relief should pursue claims for infringement of the audiovisual displays or nonliteral code similarities.

In addition, additional parameters for legal reverse engineering need to be established. In *Sega*, the Ninth Circuit delineated some practical limits which allow dissemination of ideas, but prohibit wholesale piracy.²⁹⁷ Reverse engineering should be allowed only to grasp a program's underlying functions and ideas, not to appropriate the code itself. Hopefully, the other circuits will adopt and refine *Sega*'s guidelines.

291. See *supra* notes 150-55, 174, 212-13 and accompanying text.

292. Levi, *supra* note 116, at 3; Fortnow, *supra* note 219, at 433.

293. See *supra* note 219.

294. 706 F. Supp. 984 (D. Conn. 1989).

295. *Id.* at 993.

296. Other commentators advocate that the "look and feel" analysis should not be abandoned, but employed only for very artistic works, not functional expression such as computer programs. Fortnow, *supra* note 219, at 426-27.

297. *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992); see *supra* note 103 and accompanying text.

C. *Alternative Forms of Intellectual Property Protection*

If a programmer has a truly novel idea that she wishes to protect, she has the opportunity to patent it.²⁹⁸ Although copyright law provides relatively "quick and economical protection"²⁹⁹ against flagrant copying, proof of substantial similarity can be arduous and expensive. To establish patent infringement, the owner need not prove copying or substantial similarity, but only that the infringing program falls within the patent's claims.³⁰⁰ However, a patent is more expensive and takes longer to obtain than a copyright.³⁰¹

Despite the Copyright Office's preference for single registrations, programmers who wish to copyright the unique structure and sequence of screen displays should register them as pictorial or graphic works³⁰² or audiovisual works. Because copyright law protects audiovisual works as a "series of related images,"³⁰³ the required level of creativity for copyright protection "is met by either the individual screens or the relationship of each screen to the others"³⁰⁴ This illustrates a key flaw in a pure analytical dissection analysis: failure to consider the arrangement of unprotected elements. The test proposed by this Note analyzes both the elements and their arrangement, thus protecting a programmer's rights in individual screen displays and their sequence.

To complement copyright protection, programmers may also rely on state trade secret laws.³⁰⁵ Because proving copyright

298. Patent protection requires the programmer to meet a higher standard of creativity, "nonobviousness," than the low threshold of minimal creativity required for copyright protection. 35 U.S.C. § 103 (1988). McCabe & Tanenbaum, *supra* note 36, at 4. However, patent protection gives the programmer a monopoly on the new technique. Given the recent erosion of copyright protection, one commentator advocates that patent registration should be a programmer's primary protection. *Id.*

299. McCabe & Tanenbaum, *supra* note 36, at 2. The plaintiff asserting copyright infringement must only prove a likelihood of success on the merits to win a preliminary injunction, which may be enough to force a settlement (license agreement or royalties) or put a competitor out of business before trial.

300. *Id.* at 4.

301. *Id.* Copyright attaches once a work is fixed in a tangible medium of expression. 17 U.S.C. § 102 (1988).

302. 17 U.S.C. § 101 (1988); see Fortnow, *supra* note 219, at 435.

303. 17 U.S.C. § 101 (1988).

304. Atari Games Corp. v. Oman, 979 F.2d 242, 244 (D.C. Cir. 1992).

305. Licensing agreements based on trade secret law may outlast copyright-based licensing agreements, so long as the secret remains undisclosed. Ignatin, *supra* note 159, at 2016-19. Patenting software requires full disclosure to the Patent Office, thus

infringement and misappropriation of a trade secret often require the plaintiff to prove the same elements, the federal copyright action may preempt the state trade secret claim.³⁰⁶ However, state trade secret laws are not preempted by the Copyright Act if state claims require different or additional elements of proof or create rights exclusive of those created by the Act.³⁰⁷ Finally, a trade secret may be discovered via reverse engineering limiting the usefulness of trade secret protection.

Federal trade dress laws may also protect distinctive user interfaces.³⁰⁸ To warrant trade dress protection, the interface must be inherently distinctive or have acquired secondary meaning.³⁰⁹ Additionally, the dress must be nonfunctional and the infringing trade dress must create a likelihood of confusion as to the source.³¹⁰

precluding trade secret protection. 35 U.S.C. §§ 111-112 (1988).

306. 17 U.S.C. § 301(a) (1988) expressly preempts

all legal or equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright as specified in section 106 in works of authorship that are fixed in a tangible medium of expression and come within the subject matter of copyright as specified by sections 102 and 103

However, the broad scope of preemption expressly excludes "activities violating legal or equitable rights *that are not equivalent* to any of the exclusive rights within the general scope of copyright as specified by section 106" 17 U.S.C. § 301(b)(3) (1988) (emphasis added).

307. The Eighth Circuit articulated a two-part test to determine if a state action is preempted by the Copyright Act. The state action is preempted if (1) the work is within the scope of the Copyright Act, 17 U.S.C. §§ 102-103, and (2) the state-created right is equivalent to the copyright holder's exclusive rights as enumerated in 17 U.S.C. § 106. *National Car Rental Sys., Inc. v. Computer Assoc. Int'l, Inc.*, 991 F.2d 426, 428-29 (8th Cir. 1993); see *Trandes Corp. v. Guy F. Atkinson Co.*, 996 F.2d 655 (4th Cir. 1993) (holding that misappropriation of trade secret claim not preempted); *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 716-17 (2d Cir. 1992) (holding that "extra element" must create *qualitative* change from copyright infringement claim); *CMAX/Cleveland, Inc. v. UCR, Inc.*, 804 F. Supp. 337, 358-59 (M.D. Ga. 1992). In *Computer Assoc.*, the Second Circuit first affirmed the district court's holding that the Copyright Act preempted CA's state trade secret claim, then reversed and remanded in an amended opinion. 982 F.2d at 715-16.

308. Trade dress is governed by Section 43(a) of the Lanham Act, codified at 15 U.S.C. § 1125(a) (1988).

309. "The user interface of a computer program, especially a graphical user interface, may create a sufficiently distinct visual impression, when taken as a whole, to be 'inherently distinctive.'" David L. Hayes, *What's Left of "Look and Feel": A Current Analysis* (pt. 3), *COMPUTER LAW.*, July 1993, at 13. If not inherently distinctive, the interface may acquire "secondary meaning" when the public associates a particular trade dress with its source. *Id.*

310. Hayes, *infra* note 309, at 14. Although the individual elements may be functional, their distinctive visual arrangement as a whole may be nonfunctional, and

Copyright owners may also pursue traditional state contractual remedies for breach of licensing or employment agreements.³¹¹ If the owner's rights arise from a contractual agreement, not from the Copyright Act, the state claim may not be preempted.³¹² Finally, an owner may also bring state deceptive or unfair trade practices claims where appropriate.³¹³

CONCLUSION

Recent decisions have eroded the scope of copyright protection of software, but have done so to further the constitutional mandate to promote progress by shielding only protected expression and allowing ideas to flow freely.³¹⁴ "The copyright holder has the property interest in preventing others from reaping the fruits of his original expression, not in preventing the authors and thinkers of the future from making use of, or building upon, his advances."³¹⁵ Because copyright protection is so easy to obtain and lasts for so long,³¹⁶ courts must protect only that which is truly creative expression. Overprotection limits socially beneficial and constitutionally directed innovation.³¹⁷

Copyright protection should not be used as a shield to prevent access to expression and ideas that Congress, acting under Constitutional authority, specifically chose not to protect.³¹⁸ To do so would grant a virtual monopoly to copyright holders on both the ideas and their expression, and defeat the constitutional

therefore protectable. *Id.*

311. *National Car Rental Sys., Inc. v. Computer Assoc. Int'l, Inc.*, 991 F.2d 426 (8th Cir. 1993) (holding that breach of software licensing agreement not preempted by Copyright Act).

312. *Id.* at 432-33 ("[C]ontractual restriction on use of computer program [is] distinct from the exclusive copyright rights."); *Comprehensive Tech. Int'l, Inc. v. Software Artisans, Inc.*, 3 F.3d 730 (4th Cir. 1993) (holding that breach of covenant not to compete not preempted).

313. *Patricia Kennedy & Co., Inc. v. Zam-Cul Enter., Inc.*, 830 F. Supp. 53 (D. Mass. 1993) (holding that state deceptive trade practices claim not preempted).

314. The Supreme Court's decision in *Feist* rendering the death blow to the "sweat of the brow" doctrine is a further example of a copyright erosion trend. *McCarthy*, *supra* note 214, at 283 (citing *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991)); see *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1523 (9th Cir. 1992); *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 842 (Fed. Cir. 1992); *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 711 (2d Cir. 1992).

315. *Atari*, 975 F.2d at 843.

316. See *supra* note 35.

317. Menell, *supra* note 26, at 1082.

318. *Id.* at 1526; 17 U.S.C. § 102(b) (1988).

1994]

BYTE AFTER BYTE

587

goal of allowing others to build on these ideas.³¹⁹ “When technological change has rendered an aspect or application of the Copyright Act ambiguous, ‘the Copyright Act must be construed in light of this basic purpose.’ ”³²⁰

Disagreement and confusion among the circuits have created chaos for programmers and lawyers trying to predict the future. Without guidance, software developers will continue to waste valuable resources litigating instead of innovating. Until the circuits agree, or the Supreme Court defines uniform, workable standards, the confusion will persist.

Virginia Ann Johnson

319. U.S. CONST. art. I, § 8, cl. 8.

320. *Twentieth Century Music Corp. v. Aiken*, 422 U.S. 151, 156 (1975), *quoted in* *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1527 (9th Cir. 1992).